

FFI RAPPORT

SYNTETISK MODELLERING AV RADARREFLEKSJONER I FPGA DESIGN

THINGSRUD Øyvind

FFI/RAPPORT-2004/03047

**SYNTETISK MODELLERING AV
RADARREFLEKSJONER I FPGA DESIGN**

THINGSRUD Øyvind

FFI/RAPPORT-2004/03047

FORSVARETS FORSKNINGSINSTITUTT
Norwegian Defence Research Establishment
Postboks 25, 2027 Kjeller, Norge

P O BOX 25
 NO-2027 KJELLER, NORWAY
REPORT DOCUMENTATION PAGE

SECURITY CLASSIFICATION OF THIS PAGE
 (when data entered)

1) PUBL/REPORT NUMBER FFI/RAPPORT-2004/03047 1a) PROJECT REFERENCE FFI-II/838/915	2) SECURITY CLASSIFICATION UNCLASSIFIED 2a) DECLASSIFICATION/DOWNGRADING SCHEDULE -	3) NUMBER OF PAGES 125		
4) TITLE SYNTETISK MODELLERING AV RADARREFLEKSJONER I FPGA DESIGN SYNTHETIC MODELLING OF RADAR REFLECTIONS IN FPGA DESIGN				
5) NAMES OF AUTHOR(S) IN FULL (surname first) THINGSRUD Øyvind				
6) DISTRIBUTION STATEMENT Approved for public release. Distribution unlimited. (Offentlig tilgjengelig)				
7) INDEXING TERMS IN ENGLISH: <table style="width: 100%; border: none;"> <tr> <td style="width: 50%; vertical-align: top;"> a) <u>Electronic Warfare</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>High-resolution radar</u> e) <u>DRFM-modulator design</u> </td> <td style="width: 50%; vertical-align: top;"> IN NORWEGIAN: a) <u>Elektronisk Krigføring</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>Høyoppløselig radar</u> e) <u>DRFM modulatordesign</u> </td> </tr> </table>			a) <u>Electronic Warfare</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>High-resolution radar</u> e) <u>DRFM-modulator design</u>	IN NORWEGIAN: a) <u>Elektronisk Krigføring</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>Høyoppløselig radar</u> e) <u>DRFM modulatordesign</u>
a) <u>Electronic Warfare</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>High-resolution radar</u> e) <u>DRFM-modulator design</u>	IN NORWEGIAN: a) <u>Elektronisk Krigføring</u> b) <u>Jamming</u> c) <u>DRFM</u> d) <u>Høyoppløselig radar</u> e) <u>DRFM modulatordesign</u>			
THESAURUS REFERENCE: 8) ABSTRACT <p>This thesis for the cand scient degree at the University of Oslo at Department of Informatics is documenting work done on synthetic modelling of radar reflections and concepts for implementation in an experimental radar jammer at the Norwegian Defence Research Establishment (FFI).</p> <p>The thesis starts with radar and radar jamming history, before an introduction to the fundamentals on radar systems, especially on digital pulse compression. Then today's radar jammers and the Digital Radio Frequency Memory (DRFM) are introduced. Different concepts for synthetic modelling of radar reflections are simulated and verified with the software tool MatLab, before possible realisation structures for the jammer modulator are analysed. At the end of the thesis possible implementations and hardware technologies for the modulator are worked out. The final conclusion on the thesis is that the requested jammer modulator can be implemented in a multi-chip Field Programmable Gate Array (FPGA) structure and can be configured to handle a number of situations and tasks.</p>				
9) DATE 2004-09-13	AUTHORIZED BY This page only Vidar Andersen	POSITION Director		

INNHOLD

	Side
1 INNLEDNING	7
APPENDIKS	
A HOVEDFAGSAVHANDLING	8

SYNTETISK MODELLERING AV RADARREFLEKSJONER I FPGA DESIGN

1 INNLEDNING

Denne rapporten inneholder en hovedfagsavhandling med tittel ”*Syntetisk modellering av radarrefleksjoner i FPGA design*” og med undertittel ”*En studie av DRFM modulatordesign for høyoppløselig radarjammer*”.

Avhandlingen er skrevet som en del av et hovedfagsstudium til cand scient graden ved Universitetet i Oslo, Institutt for informatikk, under veiledning av førsteamanuensis Jim Tørresen, Ifi, UiO og forsker Stein Kristoffersen, FFI.

Rapporten omhandler konstruksjon av digital radio frekvens minne (DRFM) og spesielt dens modulator ved bruk mot høyoppløselige radarer. Avhandlingen beskriver en alternativ måte for konstruksjon av denne modulatoren i forhold til dagens design i FFIs eksperimentelle radarjammer EKKO II.

Hovedfagsavhandlingen er i sin helhet vedlagt i appendiks A.

APPENDIKS

A HOVEDFAGSAVHANDLING

Syntetisk modellering av radarrefleksjoner i FPGA design

*En studie av DRFM modulatordesign
for høyoppløselig radarjammer*

Øyvind Thingsrud

**Institutt for informatikk
Universitetet i Oslo**

25 juli 2004



Forord

Jeg vil rette en takk til Forsvarets forskningsinstitutt (FFI) som har gitt meg anledning til å gå litt dypere i fagfeltet radar og dens signalbehandling – sett i fra et elektronisk krigføringsperspektiv. Jeg har studert problemstillinger rundt syntetisk generering av komplekse radarrefleksjoner og tilhørende sanntids implementasjoner i *Field Programmable Gate Array* (FPGA) teknologi.

Jeg vil spesielt takke prosjektleder Tor-Odd Høydal, FFI og tidligere prosjektleder Arne-Petter Bartholsen, FFI som har gjort dette hovedfagsstudiet mulig.

Jeg ønsker også å takke mine veiledere Stein Kristoffersen, FFI og Jim Tørresen, Ifi, UiO, og alle mine kolleger ved FFI – spesielt Tor Holmboe, og min familie for deres uvurderlige hjelp, støtte og tålmodighet under mitt hovedfagsarbeid.

Abstract

This thesis for the cand scient degree at the University of Oslo at Department of Informatics is documenting work done on synthetic modelling of radar reflections and concepts for implementation in an experimental radar jammer at the Norwegian Defence Research Establishment (FFI).

The thesis starts with radar and radar jamming history, before an introduction to the fundamentals on radar systems, especially on digital pulse compression. Then today's radar jammers and the Digital Radio Frequency Memory (DRFM) are introduced.

Different concepts for synthetic modelling of radar reflections are simulated and verified with the software tool MatLab, before possible realisation structures for the jammer modulator are analysed. At the end of the thesis possible implementations and hardware technologies for the modulator are worked out.

The final conclusion on the thesis is that the requested jammer modulator can be implemented in a multi-chip Field Programmable Gate Array (FPGA) structure and can be configured to handle a number of situations and tasks.

Innhold

FORORD	III
ABSTRACT	V
INNHold	VII
1. INNLEDNING	1
1.1 OPPGAVENS PROBLEMSTILLING.....	2
1.2 BAKGRUNNEN FOR ARBEIDENE.....	3
1.3 MÅLSETTINGEN FOR ARBEIDENE	3
1.4 ORGANISERINGEN AV HOVEDFAGSSTUDIET OG AVHANDLINGEN	4
2. LITT HISTORIE OM RADAR OG JAMMERUTVIKLING	6
2.1 RADARENS UTVIKLING	6
2.2 PASSIVE MOTMIDLER MOT RADAR	7
2.3 AKTIVE MOTMIDLER MOT RADAR	7
3. RADARTEORI	9
3.1 GRUNNLEGGENDE PRINSIPPER	9
3.1.1 <i>Forholdet tidsforsinkelse og avstand</i>	9
3.1.2 <i>Dopplerskift</i>	10
3.1.3 <i>Radarligningen</i>	10
3.1.4 <i>Radartverrsnitt</i>	11
3.1.5 <i>Multiple radarrefleksjoner</i>	12
3.2 PULSKOMPRESJON, BØLGEFORM OG TID-BÅNDBREDDE-PRODUKT	13
3.2.1 <i>Lineær frekvensmodulasjon</i>	13
3.2.2 <i>Fasemodulasjon</i>	15
3.2.2.1 <i>Optimale fasekoder</i>	15
3.2.2.2 <i>Maksimallengde fasekoder</i>	15
3.3 PROSESSERING AV KOHERENTE PULSER.....	16
3.4 EKSEMPLER PÅ RADARER SOM BENYTTER KOHERENT PROSESSERING.....	17
3.4.1 <i>High-Range-Resolution Radar (HRR)</i>	17

3.4.2 Synthetic Aperture Radar (SAR).....	18
3.4.3 Inverse Synthetic Aperture Radar (ISAR).....	20
4. DIGITALT RADIO FREKVENNS MINNE (DRFM) TEKNOLOGI	21
4.1 INNLEDNING	21
4.2 BLOKKSJEMATISK BESKRIVELSE.....	21
4.3 MIKROBØLGE OPP/NED-KONVERTER.....	22
4.4 DRFM-KJERNE.....	24
4.4.1 Analog-til-digital konverterne (A/D).....	24
4.4.2 Datahukommelse	26
4.4.3 Modulator	27
4.4.4 Digital-til-analog konverterne (D/A).....	29
4.5 TEKNIKKGENERATOR	30
4.6 EKKO II – EKSPERIMENTELL DRFM RADARJAMMER	31
5. MODELLERING AV RADARREFLEKSJONER	34
5.1 INNLEDNING	34
5.2 AKTUELLE METODER FOR RADARJAMMER MODELLERING	34
5.3 DATAMODELL AV DRFM-MODULATOR OG ENKELTE RADARFUNKSJONER	38
5.4 MODUL FOR GENERERING AV RADARBØLGEFORMER	38
5.4.1 Chirp.....	39
5.4.2 Fasekodede pulser	39
5.5 MODUL FOR PULSKOMPRESJON I RADARMOTTAKER	41
5.6 MODUL FOR DRFM-MODULATOR	42
5.7 DEFINISJON AV OBJEKT MED ROMLIG UTSTREKNING FOR TEST AV DATAMODELLEN	44
5.8 EKSEMPEL PÅ MODELLERING AV RADARREFLEKSJONER FOR HRR-RADAR	46
6. ANALYSE AV MULIGE REALISERINGSKONSEPTER FOR MODULATOREN.....	49
6.1 INNLEDNING	49
6.2 KRAV FOR VALG AV REALISERINGSMETODE FOR DIGITAL MODULATOR.....	49
6.3 DEFINISJON AV REFERANSE SIGNAL.....	50
6.4 REFERANSESTRUKTUR – DIREKTE REALISERING	50
6.5 STANDARD KONVOLUSJON.....	52

6.6 FCP-KONVOLUSJON	53
6.7 FCP-KONVOLUSJON MED SEPARAT MODULASJON AV OBJEKT OG BAKGRUNN	56
6.8 PARALLELLE FCP-KONVOLUSJONER	57
6.9 DISTRIBUERTE FCP-KONVOLUSJONER	61
6.10 DISTRIBUERTE FCP-KONV. MED SEPARAT MODULASJON AV OBJEKT OG BAKGRUNN	62
6.11 DISTRIBUERT FCP-KONV. MED PARAMETRISK GENERATOR FOR BAKGRUNNSPROFIL	64
6.11.1 Parametrisk generering av bakgrunnsprofil	65
6.11.2 Databuffere mellom høyhastighetslogikk og modulasjonslogikk	67
6.12 EKSEMPEL I PÅ VALG AV REALISERINGSSTRUKTUR	67
6.12.1 Standard konvolusjon	68
6.12.2 FCP-konvolusjon	68
6.12.3 FCP-konvolusjon med separat modellering av objekt og bakgrunn	69
6.12.4 Parallell / distribuerte FCP-konvolusjoner	69
6.12.5 Distribuert FCP-konv. med separat modulasjon, evt. med parametrisk generator	70
6.12.6 Sammendrag av realiseringsforslag for eksempel I	70
6.13 EKSEMPEL II PÅ VALG AV REALISERINGSSTRUKTUR	70
6.14 DISKUSJON	73
6.14.1 Standard konvolusjon	73
6.14.2 FCP-konvolusjon	74
6.14.3 FCP-konvolusjon med separat modellering av objekt og bakgrunn	74
6.14.4 Parallell FCP-konvolusjoner	74
6.14.5 Distribuerte FCP-konvolusjoner	75
6.14.6 Distribuert FCP-konvolusjon med separat modulasjon av objekt og bakgrunn	75
6.14.7 Distribuert FCP-konvolusjon med parametrisk generator for bakgrunnsprofil	75
6.15 KONKLUSJON	75
7. ANALYSE AV DIGITALISERINGEN I EN DRFM BRUKT MOT HRR-RADAR.....	77
7.1 INNLEDNING	77
7.2 HVA DIGITALISERINGEN GJØR MED SIGNALENE	77
7.3 KOMPLEKSITETEN I GENERERT JAMMESIGNAL MOT HØYOPPLØSELIG RADAR	78
7.4 DEFINISJON AV REFERANSE SIGNAL.....	79

7.5 MÅL FOR SIGNALENES 'GODHET' ETTER DIGITALISERING, MODULERING OG KOMPRIMERING....	80
7.6 GJENNOMFØRINGEN AV ANALYSEN	80
7.7 TESTRESULTATER	81
7.7.1 A/D-konvertere	81
7.7.2 Modulatorens amplitudemodulasjon	82
7.7.3 Modulatorens addisjonsnettverk og D/A-konvertere.....	83
7.8 DISKUSJON	84
7.8.1 A/D-konvertere	84
7.8.2 Referansesignalene.....	84
7.8.3 Modulatorens amplitudemodulasjon	85
7.8.4 Modulatorens addisjonsnettverk og D/A-konvertere.....	85
7.9 KONKLUSJON.....	85
8. ANALYSE AV MULIGE MODULATORIMPLEMENTASJONER FOR EKKO II	86
8.1 INNLEDNING	86
8.2 MODULATORENS STRUKTURELLE OPPBYGNING I EKKO II.....	86
8.3 VALG AV FPGA FOR MODULATORKORTENE.....	87
8.4 MODULATORKORTETS OPPBYGNING.....	89
8.4.1 Direkte realisert modellering med foreslått FPGA-struktur	94
8.4.2 Beregnet realisert modellering med foreslått FPGA-struktur	95
8.5 IP-KJERNER FOR FFT / IFFT PROSESSERING	97
8.5.1 Xilinx Inc – CoreGen.....	98
8.5.2 Pentek Inc – GateFlow	98
8.5.3 Roke Manor Research	99
8.5.4 RF Engines Limited – Vectis QuadSpeed	100
8.6 KONKLUSJON.....	101
9. KONKLUSJON OG VIDERE ARBEIDER.....	103
KILDELISTE	105

1. Innledning

Denne hovedfagsoppgaven omhandler syntetisk modellering av radarrefleksjoner i en *modulator* som inngår i en radarjammer under utvikling ved Forsvarets forskningsinstitutt (FFI). Modulatoren skal sørge for å modulere radarsignalene på en slik måte at refleksjonene blir presentert som troverdige mål på et radarbilde. Før selve oppgavens problemstilling blir gjennomgått, vil sentrale begreper som er nødvendige for å forstå sammenhengen bli introdusert.

En radar er en innretning som ved hjelp av elektromagnetiske signaler kan overvåke et visst område og tegne et bilde av dette på en radarskjerm, ref [6]. Radar kan benyttes i bakkeinstallasjoner, på land-, luft- eller sjø-farkoster, og kan avbilde landområder, bygninger, farkoster og endog personell. Radarens prinsipp bygger på utsendelse av et elektromagnetisk signal, for så å observere de refleksjonene som genereres av hindringer og andre objekter i terrenget. Normalt benytter radarer sendefrekvenser i GHz-området, mens signalbehandlingen av refleksjonene foregår i basebånd etter en frekvenskonvertering. Typisk ligger basebåndet i MHz-området.

En radarjammer er utstyr for å hindre en radars innhenting av informasjon, og baserer seg som radaren på utsending av elektromagnetisk energi, ref kapittel 2.3 og [12]. Støyjammere sender ut energi over et stort frekvensområde, mens nyere avanserte jammere sender ut kunstige radarekko som er tilpasset de aktuelle radarsignalene. Her er det et ønske om å generere troverdige radarmål med oppførsel som andre mål, alt for å lage forvirring for den som betrakter radarbildet. Dette gjøres ved at ekte radarpulser blir 'tatt opp' ved hjelp av en digital lagringsenhet, manipuleres og så gjenutsendes som jamming.

Et digitalt radio frekvens minne (DRFM), ref kapittel 4 og [15], er en digital lagringsenhet som gjør opptak av radarsignaler og tillater manipulasjon av en eller flere av radarsignalets parametere før gjenutsending. DRFM er en nøkkelkomponent i alle nyere radarjammesystemer. Manipulasjonen som blir utført i DRFM-modulatoren kan opptre som endrede målinger av avstand og hastighet til radarmålet, og endog flere kopier av radarmålet med forskjellig avstander og hastigheter kan genereres. Dette kalles generering av falske radarmål.

Når avansert manipulasjon utføres blir dette ofte kalt syntetisk modellering av radarrefleksjoner, ref kapittel 5 og [3]. Denne type modellering gjøres helst mot høyoppløselige radarer, dvs radarer med svært god avstandsoppløsning. Modelleringen har her som målsetting å generere troverdige refleksjoner tilsvarende refleksjonene fra f.eks komplekse menneskelagde objekter med et mangfold av refleksjonspunkter.

Det er i dag flere typer høyoppløselige radarer; *High-Range-Resolution Radar* (HRR), *Synthetic Aperture Radar* (SAR) og *Inverse Synthetic Aperture Radar* (ISAR) er de mest kjente, ref [7]. Alle disse radarene baserer seg på mye av de samme prinsippene og derfor vil syntetisk modellering av radarrefleksjoner for disse systemene også ha mye til felles. Denne avhandlingen har dog kun fokus på modellering av radarrefleksjoner for HRR-radar.

Alle høyoppløselige radarer har egenskaper for å avbilde objekter og/eller terreng med deres karakteristiske egenskaper. Grunnene for avbildningen eller overvåkingen kan være av

miljømessig karakter eller være en viktig del av naturressursforvaltningen, eller være rent militære. Høyoppløselige radarer utfører avbildning i 1, 2 eller 3 dimensjoner. Betegnelsen avbildning benyttes selv om kun 1-dimensjonal avbildning foretas. 1-dimensjonal avbildning genererer en avstandsprofil av objektet, mens 2 og 3-dimensjonale avbildninger gir bilder som i enkelte radarmodi til en viss grad kan ligne på fotografiske bilder.

I ett av de få eksperimentelle DRFM-systemene som kan utføre syntetisk modellering, har dette blitt realisert ved hjelp av digital signalbehandling og direkte modellering, ref [2]. Direkte modellering er en metode hvor man konstruerer forsinkelse, frekvens og amplitudeforandring ved hjelp av elektroniske komponenter – altså en direkte etterligning av den fysiske refleksjonsprosessen i naturen. I stedet for å bygge opp konstruksjonen ved hjelp av diskrete komponenter, implementeres dette gjerne digitalt og integreres i *Field Programmable Gate Arrays* (FPGA) som tillater store mengder digital logikk i en reprogrammerbar krets.

Tema for denne avhandlingen er nettopp syntetisk modellering av komplekse radarrefleksjoner implementert i FPGA og hvordan dette kan gjøres på en annerledes måte enn slik det til nå har vært gjort.

1.1 Oppgavens problemstilling

En alternativ fremgangsmåte for syntetisk modellering av radarrefleksjoner og realisering i en radarjammers DRFM-modulator skal utvikles. Denne metoden benytter digital signalbehandling og er prinsipielt beskrevet i [1], men er foreløpig ikke blitt demonstrert i bruk. Før dette kan skje må en rekke problemstillinger avklares. Metoden blir i [1] omtalt som *computed* modellering, heretter kalt beregnet modellering på norsk i denne avhandlingen.

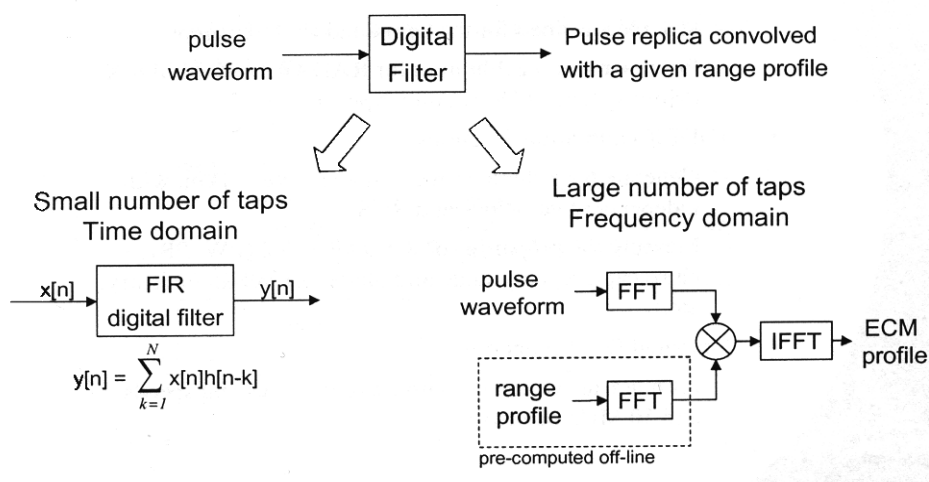
Syntetisk modellering kan dermed enten realiseres ved hjelp av direkte modellering eller ved hjelp av beregnet modellering. Fordelen med dagens direkte modellering er at metoden er lett å forstå, samtidig som innskuddsforsinkelsen gjennom modulatorene er såpass liten at dette ikke får uheldige bivirkninger for radarrefleksjonenes troverdighet. Ulempene med dagens direkte metode er at maskinvaren fort blir ressurskrevende når store antall radarrefleksjoner skal modelleres, ref [1].

Av denne grunn er det stor interesse for å erstatte denne metoden med beregnet modellering hvor modulasjonsprosessen ses på som et digitalt filter. Denne metoden gjør at kompleksiteten blir mer uavhengig av antall refleksjoner som skal genereres. Her kan man enten modellere radarrefleksjonene i tidsdomenet eller frekvensdomenet. For transformasjon mellom tid og frekvensdomenet, og tilbake igjen, benyttes henholdsvis *Fast Fourier Transform* (FFT) og invers FFT. Selve modulasjonen må utføres ved hjelp av kompleks konvolusjon i tidsdomenet eller med kompleks multiplikasjon i frekvensdomenet. Beregningene som inngår i en kompleks konvolusjon er meget regnekrevende, mens komplekse FFT-transformasjoner og komplekse multiplikasjoner er mindre krevende.

I alle tilfeller kreves en effektiv realisering, sett i lys av den relativt korte beregningstiden som er tilgjengelig. Akseptabel beregningstid er for de fleste radarsystemer lik radarens *Pulse Repetition Interval* (PRI), dvs tiden mellom hver radarutsendelse. På denne måten vil

et mottatt radarsignal kunne benyttes til beregning av refleksjonene som skal brukes til jamming av neste utsendte radarsignal.

De to aktuelle måtene å foreta beregnet modellering på, er altså i tidsdomenet eller i frekvensdomenet. Ved et relativt lite antall refleksjonspunkter vil en digital filterløsning i tidsdomenet være mest hensiktsmessig, mens ved store antall refleksjoner vil konvertering til frekvensdomenet, for så å foreta modelleringen der kunne lønne seg, ref [1]. Figur 1.1 viser skjematisk hvordan modelleringen er tenkt.



Figur 1.1 Beregnet modellering i henholdsvis tidsdomenet og frekvensdomenet, ill. fra [1]

1.2 Bakgrunnen for arbeidene

FFI arbeider for tiden med utvikling av en ny eksperimentell radarjammer med navn *EKKO II*, ref [2] og [3]. Den er basert på DRFM-teknologi og skal benyttes ved utvikling av jammekonsepser mot blant annet høyoppløselige radarer som HRR, SAR og ISAR.

FFI har i en årrekke utviklet flere forskjellige jammesystemer både for operativ bruk og for eksperimentelt bruk, da *Elektronisk Krigføring* (EK) som dette fagfeltet kalles, har vært og er et satsningsområde ved FFI. Radarjammen *EKKO II* er en oppfølger til den eksperimentelle jammen *EKKO I*, men vil få en langt mer utvidet funksjonalitet og fleksibilitet enn forgjengeren, noe som er spesielt viktig under forsøk og annet eksperimentelt bruk.

1.3 Målsettingen for arbeidene

Denne hovedfagsoppgaven har som målsetting å utvikle et konsept for en DRFM-modulator til radarjammen *EKKO II*. Modulatoren skal ved hjelp av beregnet realisert modellering kunne modellere et stort antall radarrefleksjoner.

Det skal utvikles en datamodell som modellerer et høyoppløselig radarsignals refleksjon fra et objekt med romlig utstrekning. Det er denne refleksjonen som skal reproduseres i jammerens DRFM-modulator. Videre skal det datamodelleres hvordan radarmottakerens signalprosessering håndterer det kunstig genererte jammesignalet.

Det skal utvikles forskjellige strukturer for DRFM-modulatoren med beregnet realisert modellering som kan sammenlignes mot hverandre og mot dagens benyttede direkte modellerte realisering. De enkelte strukturene kan velges avhengig av aktuell radartype som skal håndteres, da modulatorens oppbygning kan varieres etter behov ved hjelp av programmeringen av FPGA-kretsene som modulatorens er forutsatt å skulle bygges opp av.

Deretter skal egnede FPGA-kretser vurderes og foreslås, og en struktur for sammenkobling av flere FPGA-kretser for modulatorene skal utvikles. Denne koblingsstrukturen skal tillate implementasjon av både dagens direkte modellerte realisering og den kommende beregnede modellerte realisering.

1.4 Organiseringen av hovedfagsstudiet og avhandlingen

Arbeidet vedrørende syntetisk modellering av radarrefleksjoner har blitt utført ved FFI som en forskningsoppgave under et hovedfagsstudium ved Universitetet i Oslo, Institutt for informatikk, innen studieretning digitalteknikk.

I studiet inngår et teoretisk pensum hvor 10 vekttall er studert. Det består av 4 emner, ref Tabell 1.1, som dels har tilknytning til fagfeltet og dels gir studiet større faglig bredde. I tillegg har deltagelse på konferansen EUSAR 2004 vært en del av studiet, ref [23], som andreforfatter av poster sammen med Stein Kristoffersen, FFI.

Emnekode	Emne	vektall
UNIKF 361	Radar	3
UNIKF-ABIR	Avbildende radar	3
UNIKI 322	Overvåkningsteori	2
UNIKFI-HDIK	Høynivåkonstruksjon av digitale integrerte kretser	3

Tabell 1.1 Hovedfagsstudiets teoretiske pensum

Avhandlingen er organisert på følgende måte:

Kapittel 1 gir en innledning i hva hovedfagsstudiet har omfattet.

Kapittel 2 tar for seg litt historikk om radar og radarjamming.

Kapittel 3 gir en kort innføring i radarteori. Først det helt grunnleggende, deretter en gjennomgang av pulskompresjonens egenskaper, forskjellige bølgeformer som benyttes ved høyoppløselig radar og uttrykket tid-båndbredde-produkt. Dernest en kort omtale av HRR, SAR og ISAR-radarene.

Kapittel 4 gir en gjennomgang og en del betraktninger rundt oppbygningen av et moderne DRFM-system med FFIs EKKO II som eksempel.

Kapittel 5 er avhandlingens første sentrale kapittel, med gjennomgang av aktuelle metoder for syntetisk modellering av radarrefleksjoner, og beskrivelse av datamodelleringens forskjellige moduler; modul for generering av radarbølgeformer, modul for pulskompresjon i radarmottaker og modul for DRFM-modulatorens funksjon. Datamodelleringsprogrammet MatLab har blitt benyttet til dette. Resultatene fra datamodelleringen blir presentert med et eksempel som viser modellering av radarrefleksjoner for HRR-radar. Datamodellen blir også benyttet for visualisering og validering av resultatene i de etterfølgende analysene i avhandlingen.

Kapittel 6 er en analyse av mulige realiseringskonsepter for DRFM-modulatoren. Her blir forskjellige strukturer utviklet, og sammenlignet seg i mellom og mot referansestrukturen, direkte realisert modellering. Analysen inneholder også to eksempler for å belyse problemstillingen bedre.

Kapittel 7 er en analyse av nødvendig oppløsning i digitaliseringen hos en DRFM brukt mot høyoppløselig radar. Her fremgår de høye kravene som stilles til gjengivelse av troverdige komplekse radarrefleksjoner.

Kapittel 8 analyserer mulige implementasjonsstrukturer i FPGA for EKKO IIs DRFM-modulator. Analysen foreslår både FPGA-type og en sammenkoblingsstruktur av flere FPGA-kretser for et modulorkort som tillater implementasjon av både direkte realisert og beregnet realisert modellering.

Kapittel 9 konkluderer hele hovedfagsavhandlingen og foreslår videre arbeider vedrørende modellering av radarrefleksjoner.

Kildelisten til slutt i avhandlingen inneholder bøker, ugraderte artikler og annen informasjon om temaet, blant annet internett referanser.

Mange av figurene som blant annet er brukt i de innledende kapitlene og enkelte av eksemplene er hentet fra forskjellige kilder. Alle 'lånte' illustrasjoner er referert med kildeanvisning. På samme måte er innhentet informasjon referanseført.

For å holde alle figurene i samme stil og for at figurene skal kunne benyttes i andre sammenhenger senere, er det valgt å ha figurene med engelsk tekst selv om avhandlingen ellers er skrevet på norsk.

2. Litt historie om radar og jammerutvikling

Det blir først gitt et tilbakeblikk på utviklingen av radar. Deretter studeres motmidlene mot radarer som er fremkommet i tidens løp.

2.1 Radarens utvikling

Radar er en forkortelse for *Radio Detection and Ranging*. Med radarens inntog kunne man 'se' i mørket og gjennom tåke. Den ble raskt et betydningsfullt verktøy innen krigføring og da i første omgang for å hindre overfallsangrep. Med tiden har radaren også blitt viktig for det sivile samfunn – spesielt innen navigasjon i luftfart og i handelsflåten, men også blant fritidsbåter.

Den skotske fysikeren James Clerk Maxwell kom i 1865, ref [17], fram til ligninger som beskriver oppførselen hos elektromagnetiske bølger og derav lovene om radiobølgerrefleksjoner. Men, først i 1888 ble det demonstrert at elektrisk ledende objekter virkelig reflekterer radiobølger av den tyske fysikeren Heinrich Hertz, ref [18], og i 1904 lagde landsmannen Christian Hülsmeier utstyret 'telemobiloscope', ref [21]. Denne enheten var en hindringsdetektor med rekkevidde 3000 m for skip som baserte seg på refleksjon av radiobølger.

De første vellykkede avstandsmålingene ved hjelp av radar ble utført i 1924 av briten Sir Edward Victor Appleton, ref [19]. Han bestemte høyden opp til ionosfæren ved hjelp av refleksjoner i dennes ioniserende lag. Dette ioniserende laget har den elektriske ledningsevnen som må være til stede for å oppnå radiobølgerrefleksjon.

Deteksjon av fly ble først gjennomført i 1930 av amerikaneren Lawrence A. Hyland, ref [20], ved Naval Research Laboratory (NRL), mens det første egentlige radarsystemet ble utviklet i 1935 av briten Sir Robert Watson-Watt, ref [21]. I løpet av 1939 var en kjede av disse radarsystemene etablert langs Englands sør- og østkyst. Dette var pulsradarer med sendefrekvens 22 MHz og rekkevidde opptil 150 km mot fly på 3000 m høyde. I årene rett før og under den andre verdenskrig ble flere nye radarer utviklet for beskyttelse av England, hvor høyere senderfrekvens ble benyttet og hvor antennenenes dekningsdiagram tillot overvåkning av havoverflaten ut til 20 km.

Siden den tid har utviklingen innen radar vært stor, både innen land-, fly- og rom-baserte systemer. I de senere årene har utviklingen rettet seg mot fasestyrte antenner, digital signalbehandling og avbildende radarer. Radarene SAR og ISAR gjør nytte av denne utviklingen. SAR benyttes fra fly eller satellitt for å avbilde eller overvåke et område på jorden, mens ISAR utføres for å avbilde et bevegelig objekt. Oppløsningen i SAR- og ISAR-bildene kan være under 1 meter selv ved avbildning fra svært store avstander, ref [7]. Med den rette radarprosesseringen vil bildene få en oppløsning som er uavhengig av avstand.

Etter at de første radarsystemene ble operative har det blitt arbeidet mye for å hindre at fly og andre objekter blir detektert. Rent operativt blir radarens manglende dekning i lave høyder på grunn av terrengskjerming utnyttet. Dermed kan lavtflygende fly unngå å bli

oppdaget, men denne metoden har store operative begrensninger. Derfor har flere andre konsepter, både passive og aktive blitt utviklet.

2.2 Passive motmidler mot radar

Det arbeides mye for å redusere flyenes radartverrsnitt, dvs minske mengden av radarsignalet som reflekteres tilbake til radaren. Dette kan oppnås ved konstruksjon av flyets struktur slik at radarsignalene hovedsaklig reflekteres mot andre retninger enn radarens posisjon og ved bruk av radarabsorberende materiale på flyets overflate. Dette blir ofte betegnet som '*stealth*' teknikker. Et eksempel på stealth jagerfly er omtalt i [22].

Det har også fra lang tid tilbake blitt benyttet elektroniske motmidler mot radarer – *Electronic Countermeasures* (ECM), ref [12]. Disse motmidlene har som oppgave å generere forstyrrende eller falske radarsignaler som blander seg med de ekte radarekkoene på en slik måte at radarbildet blir villedende for brukerne. I enkleste form kan dette være biter av radarreflekterende materiale som skytes ut fra plattformen som skal beskyttes. Metallfoliestrimlene som reflektorene er laget av blir kalt '*chaff*' og de er tilpasset radarsignalets bølgelengde slik at de fungerer som dipolantenner. I tidligere tider var radarfrekvensene lave og dermed måtte dipolantennene eller metallfoliestrimlene være lange. I dag hvor radarfrekvensene som regel ligger i GHz-området, er reflektorene laget av korte glassfibertråder belagt med aluminium. Chaff benyttes i dag til beskyttelse mot beskytning av både fly og skip.

2.3 Aktive motmidler mot radar

En litt mer avansert form for motmiddel er å benytte støysendere for å maskere det ekte radarekkoet, ref [12]. Slike støysendere kalles ofte for støyjammere. Når en eldre type radar mottar store mengder elektromagnetisk støy vil brukeren måtte skru ned forsterkningen for at bildet ikke skal bli helt overstyrt, dvs helt lyst uten noen detaljer. Og med lav forsterkning er ikke radaren i stand til å detektere de svake radarekkoene, som dermed forsvinner fra radarbildet. I dagens radarsystemer blir brukeren presentert et syntetisk generert digitalt radarbilde, med mye tilleggsinformasjon. Den digitale signalbehandlingen vil i mye større grad være i stand til å skille de ekte radarekkoene fra støyen. Men, i hvilken grad støysenderne reduserer radarens ytelse er avhengig av radarens grad av signalprosessering og hvilke algoritmer som benyttes.

I motsetning til støyjammere som sender ut energi over et større frekvensområde, så sender 'narrejammere' ut kunstige radarekko som er tilpasset de aktuelle radarsignalene, ref [2]. Her er det et ønske om å generere troverdige radarmål med oppførsel som andre mål, alt for å lage forvirring for den som betrakter radarbildet. Dette gjøres ved at ekte radarpulser blir 'tatt opp' ved hjelp av en digital lagringsenhet og så gjenutsendes til andre tidspunkt som jamming. Den digitale lagringsenheten kalles digitalt radio frekvens minne eller DRFM, ref kapittel 4.

Siden 1970-tallet har det blitt startet flere forsknings- og utviklingsprosjekter for å bringe frem en DRFM som virker i praksis. Selv om prinsippet er trivielt, så er de teknologiske utfordringene store. A/D-konvertering, digital lagring og D/A-konvertering av radarfrekvenser i GHz-området byr på en del problemer. Ut over 1980-tallet ble det vist flere DRFM-systemer på laboratoriene. Systemene hadde typisk 1 bit representasjon av radarpulsene og hadde begrenset båndbredde. Disse tekniske begrensningene hadde imidlertid liten avgjørende betydning, da de aktuelle radarene hadde liten båndbredde.

Rundt 1990 var flere systemer kommet i operativ bruk, og antall bits og båndbredden hadde økt. Men, fortsatt skulle 1 bits DRFM benyttes operativt i mange år til fordel for stadig høyere båndbredde. I dag finnes systemer med f.eks 3 GHz båndbredde og opptil 12 bits oppløsning, men få systemer har både høy båndbredde og høy oppløsning samtidig.

De første DRFM-systemene utførte enkle jammeteknikker for å narre operatørene eller narre signalbehandlingen i f.eks luftvernssystemer eller jagerfly. Kravene til dagens moderne jammesystemer er at også avbildende radarmodi skal kunne jammes troverdig. FFIs eksperimentelle radarjammer *EKKO II*, ref [2], er et av flere systemer som internasjonalt utvikles på dette området.

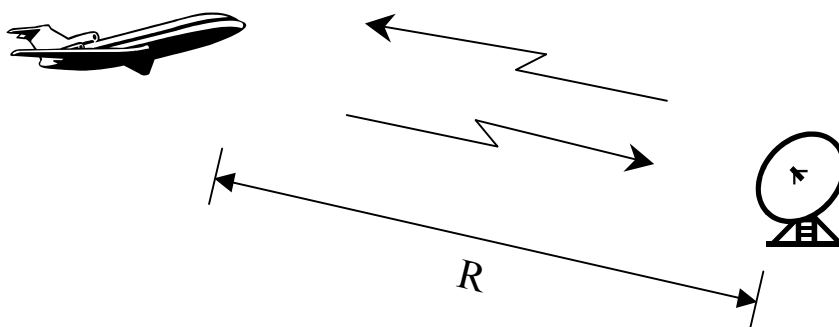
3. Radarteori

3.1 Grunnleggende prinsipper

Dette kapitlet gir en kort gjennomgang av grunnleggende prinsipper innen radarteorien som er aktuelle for denne oppgaven. Mye av tekst og illustrasjoner er hentet fra [6], [7], [8] og [9].

3.1.1 Forholdet tidsforsinkelse og avstand

En klassisk radaroppstilling består av en sender og en mottaker, og et mål ved en avstand R . Senderen utsender et elektromagnetisk signal med senterfrekvens f_c som kan være modulert i amplitude, fase eller frekvens, og gjerne i kombinasjon. Ved å sammenligne det mottatte signal med det utsendte signal, kan målets radielle avstand R fra radaren bestemmes. Antennens pekeretning i asimut og elevasjon gir målets retning i forhold til radaren.



Figur 3.1 Klassisk radaroppstilling

Forsinkelsen τ mellom utsendelses- og mottakelsestidspunktet av radarsignalet gir avstanden R til målet, ref [6]:

$$R = \frac{1}{2} \tau C_p \quad (3.1)$$

hvor C_p er propagasjonshastigheten til de elektromagnetiske bølgene i atmosfæren, og som er tilnærmet lik lyshastigheten C . Faktoren $\frac{1}{2}$ kompenserer for at tidsforsinkelsen er målt frem og tilbake til målet.

3.1.2 Dopplerskift

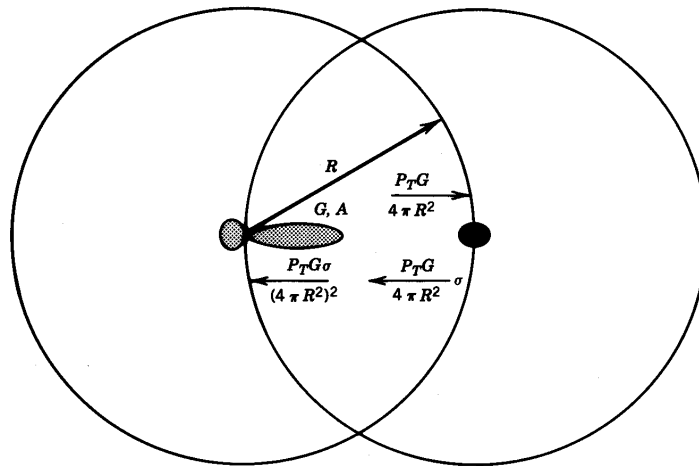
På grunn av målets radielle hastighet i forhold til radaren, vil det reflekterte radarsignalet bli frekvensskiftet i forhold til det utsendte radarsignalet. Denne frekvensdifferansen kalles dopplerskift og uttrykkes f_D , ref [6]:

$$f_D = -\frac{2v}{\lambda} \quad (3.2)$$

hvor v er målets radielle hastighet i forhold til radaren og λ er bølgelengden til radarsignalets sendefrekvens.

3.1.3 Radarligningen

Radarligningen er den fundamentale ligningen som benyttes ved dimensjonering av radarsystemer, ref [6]. Den kan utledes fra en skisse som Figur 3.2, hvor venstre sirkelsentrum angir en radar med sendereffekt P_T .



Figur 3.2 Effekttettheter i en radaroppstilling, sett ovenfra, ill. fra [6]

Ved en avstand R fra senderen vil effekttettheten bli $P_T/4\pi R^2$. Hvis senderantennen har en forsterkning G og målet har et radartverrsnitt σ , vil den reflekterte effekten fra målet bli $P_T G \sigma / 4\pi R^2$. Tilbake ved radarmottakeren vil effekttettheten da bli $P_T G \sigma / (4\pi R^2)^2$.

Gitt at mottakerantennen har effektivt areal A . Fra antenneteorien er da dens forsterkning $G = 4\pi A / \lambda^2$. Hvis dette settes inn i uttrykket for effekttettheten ved radarmottakeren fører det til den klassiske radarligningen, ref [6]:

$$S = \frac{P_T G^2 \lambda^2 \sigma}{(4\pi)^3 R^4} \quad (3.3)$$

S angir mottatt signaleffekt i radarmottakeren. I denne enkle formen av radarligningen er det sett bort fra alle tap i systemet.

Når tapene i et virkelig radarsystem tas med, vil dette føre til følgende uttrykk, ref [7]:

$$S = \frac{P_T G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L} \quad (3.4)$$

hvor S er mottatt signaleffekt i radarmottakeren og L er systemet totale tap av signaleffekt. L må være ≥ 1 i denne ligningen.

En mer nyttig form av radarligningen uttrykt ved signal-til-støyforhold $(S/N)_{in}$ tilstrekkelig for deteksjon kan utledes ved å innføre S_r i følge [7]:

$$S_r = kT_s \beta_n (S/N)_{in} \quad (3.5)$$

hvor S_r er mottakersystemets følsomhet og $kT_s \beta_n$ er termisk støy referert til mottakerens inngang. k er Boltzmanns konstant (1.38×10^{-23} W/Hz K), T_s er systemets støytemperatur og β_n er mottakersystemets støybåndbredde.

Ved å kombinere ligningene (3.4) og (3.5) kan radarligningen uttrykkes ved avstanden R gitt et visst signal-til-støyforhold $(S/N)_{in}$ referert til mottakerens inngang, ref [7]:

$$R_{max} = \sqrt[4]{\frac{P_T G^2 \lambda^2 \sigma}{(4\pi)^3 kT_s \beta_n (S/N)_{in} L}} \quad (3.6)$$

Deteksjonsavstanden R_{max} blir da den lengste avstanden radaren kan detektere et mål med radartverrsnitt σ ved det gitte signal-til-støyforhold $(S/N)_{in}$.

Hvis radarligningen settes i sammenheng med radarjamming og man tenker seg at målet i Figur 3.2 ved avstanden R har en jammer, vil radaren motta følgende jammeeffekt J_R :

$$J_R = \frac{P_J G^2 \lambda^2}{(4\pi)^2 R^2} \quad (3.7)$$

Man kan fra (3.3) se at radarens mottatte signaleffekt fra refleksjonen av målet er proporsjonal med $1/R^4$ mens mottatt jammeeffekt (3.7) er proporsjonal med $1/R^2$. Dette gir jammeren et stort fortrinn inntil avstanden R blir tilstrekkelig liten.

Et moment som det bør legges merke til er at kun den del av jammebåndbredden som dekker radarens mottakerbåndbredde vil være effektiv som jamming.

3.1.4 Radartverrsnitt

Radartverrsnittet til en kule benyttes ofte som referanse i spesifikasjon av en radars deteksjonsevne. Dvs at en kule med radartverrsnitt σ skal kunne detekteres på en viss avstand under gitte forhold. Fordelen med en kule er at radartverrsnittet er konstant for alle aspektvinkler og at den i det som kalles det optiske området er uavhengig av bølgelengden λ .

Men, generelt er radartverrsnittet til et objekt avhengig av objektets størrelse i forhold til bølgelengden λ . F.eks for en kule som er stor i forhold til bølgelengden, er radartverrsnittet i følge [6]:

$$\sigma = \pi a^2, \quad \lambda \ll a \quad (3.8)$$

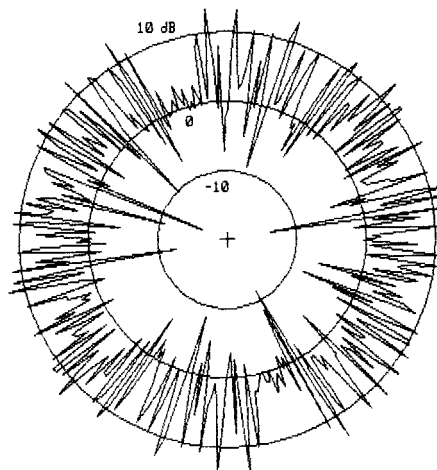
hvor a er kulens radius. Kuler som er store i forhold til bølgelengden betegnes å være i det optiske området, dvs dens radartverrsnitt er lik dens fysiske tverrsnitt. Kuler som er små i forhold til bølgelengden betegnes å være i *Rayleigh* området, og da blir radartverrsnittet i følge [6]:

$$\sigma = \pi a^2 9 \left(\frac{2\pi a}{\lambda} \right)^4, \quad \lambda \gg a \quad (3.9)$$

dvs radartverrsnittet blir vesentlig mindre enn dens fysiske størrelse skulle tilsi. Mellom det optiske området og *Rayleigh* området blir radartverrsnittet oscillatorisk pga at overflatebølger interferer med hverandre rundt kulen. Tilsvarende uttrykk som for kulen kan også finnes for forskjellige andre geometriske objekter.

3.1.5 Multiple radarrefleksjoner

Et radarekko fra et større objekt med romlig utstrekning vil typisk være sammensatt av refleksjoner fra mange enkelte refleksjonspunkter. Alle refleksjonene summerer seg vektorielt, og dermed vil kun små forandringer i reflektorenes posisjon kunne endre deres innbyrdes fase betydelig. Resultatsignalet blir derfor svært varierende i amplitude ved forskjellige aspektvinkler pga konstruktiv og destruktiv interferens. Figur 3.3 viser radartverrsnittberegninger av fem punktrefektorer for aspektvinkler i området $0 - 360^\circ$.

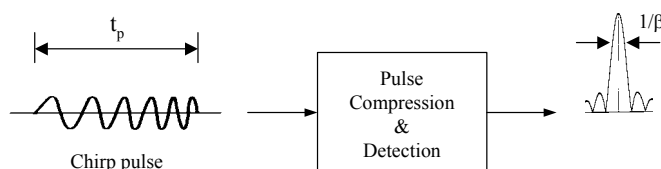


Figur 3.3 Beregnet radartverrsnitt av fem punktrefektorer, ill. fra [6]

Som Figur 3.3 viser så blir refleksjonsmønsteret svært irregulært ved varierende aspektvinkel allerede ved fem refleksjonspunkter. Nivåforskjeller i størrelsesorden 30 dB kan lett oppstå.

3.2 Pulskompresjon, bølgeform og tid-båndbredde-produkt

Pulskompresjonsradar er en radar der hvor det utsendes en lang kodet puls og den mottatte radarrefleksjonen prosesseres til å fremstå som om refleksjonen kom fra en kort radarpuls. Fordelen med pulskompresjon er at deteksjonsegenskapene forblir like gode som egenskapene til en lang puls siden utsendt energi opprettholdes, men dens egenskaper til oppløsning i avstand blir som en kort puls.



Figur 3.4 Pulskompresjon av chirp-puls

En radars avstandsopløsning Δr er definert som dens evne til å skille to objekter fra hverandre i avstand, ref [6]:

$$\Delta r = \frac{c}{2\beta} \quad (3.10)$$

hvor c er lysets hastighet og β er radarpulsens båndbredde. Hos en standard ukodet radarpuls er $\beta \approx 1/\text{pulsbredden}$, mens hos kodede pulser er β langt større.

Radarpulsene kan kodes på flere forskjellige måter for å øke dens båndbredde. En vanlig metode består i frekvens eller fasemodulasjon av utsendt puls og komprimering ved hjelp av konvolusjon mellom reflektert puls og den kompleks-konjugerte av den utsendte puls.

Betegnelsen tid-båndbredde-produkt, ofte forkortet TBW, betyr pulsens varighet multiplisert med pulsens båndbredde og er i verdi like stor som pulskompresjonsforholdet som er gitt av forholdet mellom den lange kodede puls og den korte komprimerte puls.

Pulskompresjonsforholdet eller tid-båndbredde-produktet forteller hvor mange ganger bedre avstandsopløsningen er blitt med den kodede puls kontra en ukodet puls med samme varighet.

3.2.1 Lineær frekvensmodulasjon

Lineær frekvensmodulasjon eller chirp, er den mest vanlige bølgeformen på høyoppløselige radarer. Her kan et tid-båndbredde-produkt på flere tusen oppnås. Chirp bølgeformen kan i følge [7] beskrives slik:

$$r_{ch}(t) = \text{rect}\left(\frac{t}{T_1}\right) \cdot e^{j2\pi(f_0 t + Kt^2/2)} \quad (3.11)$$

hvor

$$\text{rect}\left(\frac{t}{T_1}\right) = 1, \quad \text{hvis} \left| \frac{t}{T_1} \right| \leq \frac{1}{2}$$

$$\text{rect}\left(\frac{t}{T_1}\right) = 0, \quad \text{hvis} \left| \frac{t}{T_1} \right| > \frac{1}{2}$$

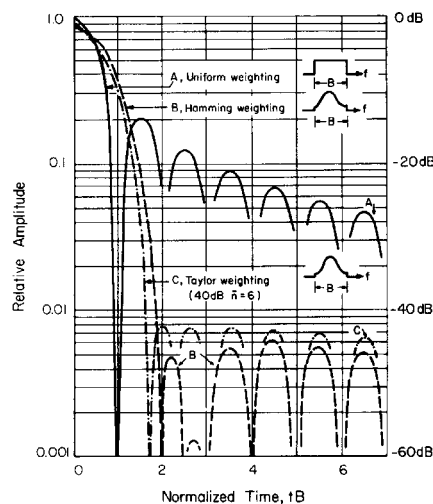
Frekvensen $f(t)$ og fasen $\phi(t)$ av $r_{ch}(t)$ er gitt av:

$$f(t) = \frac{1}{2\pi} \cdot \frac{d\phi}{dt} = f_0 + Kt$$

$$\phi(t) = 2\pi\left(f_0 t + Kt^2 / 2\right)$$

hvor den instantane frekvensen endres fra $f_0 - KT_1/2$ til $f_0 + KT_1/2$ i løpet av pulsens varighet T_1 . Chirp bølgeformen får med dette en båndbredde, $BW = KT_1$, og et tid-båndbreddeprodukt, $TBW = T_1 BW$.

Det er to ulemper med chirp bølgeformen; det ene er at det etter pulskompresjonen oppstår et relativt høyt sidelobenivå og det andre er den iboende koblingen mellom avstand og dopplerskift. Det høye sidelobenivået kan senkes til akseptable nivå ved hjelp av forskjellige vindusfunksjoner på mottatt refleksjon, se Figur 3.5.



Figur 3.5 Vindusfunksjoner for å redusere chirp sidelobenivå, ill. fra [8]

Vindusfunksjoner innfører imidlertid noe tap i signal-til-støyforholdet. I stedet for en vindusfunksjon kan en ikke-lineær chirp bølgeform benyttes.

Når det gjelder koblingen mellom avstand og hastighet, så kan dette kompenseres hvis en av de to er kjent. Ellers vil det bli introdusert en feil i målingen. Feilen består i at en endring i dopplerfrekvensen gir seg utslag som en endring i avstand og omvendt.

3.2.2 Fasemodulasjon

En annen vanlig bølgeform er fasemodulasjon eller fasekoding. Fasekoding brukes mye på radarer med midlere grad av pulskompresjon, men kan også finnes på høyoppløselige radarer.

En fasekodet bølgeform kan i følge [7] beskrives slik:

$$r_{ph}(t) = \sum_{i=0}^{N-1} b(t) \cdot e^{j(2\pi f_0 t + \phi_i)} \quad (3.12)$$

hvor

$$b(t) = 1, \quad \text{hvis } i \cdot t_1 \leq t \leq (i+1) \cdot t_1$$

$$b(t) = 0, \quad \text{ellers}$$

Pulsen $r_{ph}(t)$ har bærebølgefrequens f_0 , kodelengde N og bit-lengde t_1 . Fasen ϕ_i er enten 0 eller π avhengig av kodesekvensen.

3.2.2.1 Optimale fasekoder

En fellesbetegnelse for koder med minimale sidelobenivåer i forhold til kodelengden er optimale koder, ref [9]. De mye brukte Barker-kodene er et spesialtilfelle av de optimale fasekodene da høyeste sidelobenivå ved null doppler er $1/N$, hvor N er kodelengden.

Optimale koder er tilgjengelige for store kodelengder og kan finnes ved hjelp av dataprogrammer. De lange variantene av disse kodene kan gi et kompresjonsforhold på flere tusen ganger.

Det finnes også fasekoder hvor koden kun er optimalisert for et visst nærområde av hovedloben. På denne måten kan sidelobene undertrykkes ekstra mye der hvor det er viktigst i en gitt applikasjon.

3.2.2.2 Maksimallengde fasekoder

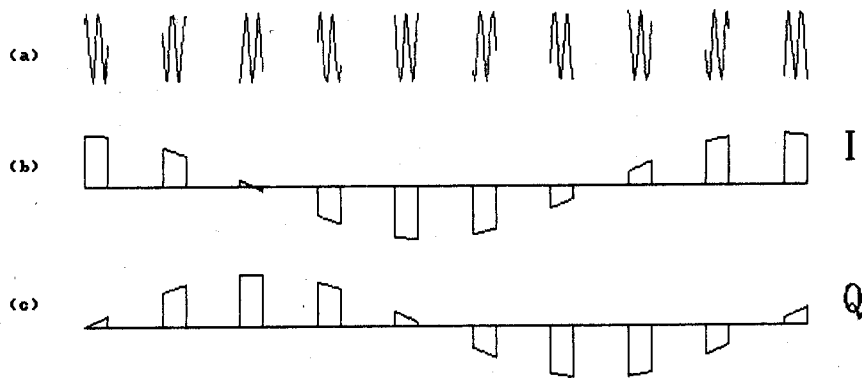
En annen variant av fasekodene er maksimallengde kodene, ref [9]. De genereres ved hjelp av et tilbakekoblet skiftregister med antall steg n og de har tilbakekobling tilpasset den enkelte kodelengde N . Disse kodene finnes for alle størrelser av n , og har kodelengder gitt av: $N = 2^n - 1$.

Maksimallengde kodene har dårligere sidelobenivå enn de optimale kodene, men er av flere grunner mye brukt allikevel. En av kodenenes egenskaper er deres pseudo-tilfeldige natur og de kalles ofte for støykode av den grunn. Et annet fortrinn er den enkle måten kodesekvensene genereres på.

3.3 Prosessering av koherente pulser

I alle nyere radartyper som HRR, SAR og ISAR er radarprosesseringen avhengig av komplekse sampler av de mottatte radarrefleksjonene. Komplekse sampler betyr at både amplitude og fase av signalet er tatt vare på, og uttrykkes ved hjelp av realdel og imaginær del av signalet. Realdel forkortes I av *in-phase* og imaginær del Q av *quadrature phase*.

Ved hjelp av I og Q sampling og deteksjon av mottatte refleksjoner kan faseinformasjonen i radarsignalet beholdes. Figur 3.6 viser et (a) pulstog med koherente pulser dvs at pulsene er utsnittet fra en kontinuerlig bærebølge, (b) I-signal etter deteksjon og (c) Q-signal etter deteksjon.



Figur 3.6 Pulstog med tilhørende I og Q-signaler, ill. fra [6]

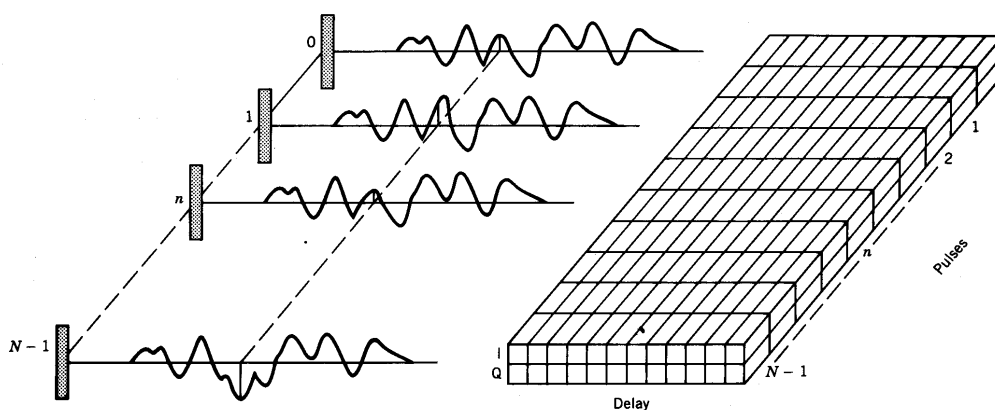
Som figuren viser inneholder I og Q-signalene faseinformasjonen fra radarsignalene og kan i følge [6] uttrykkes slik:

$$I(t) = p(t)\cos(2\pi f_D t) \quad (3.13)$$

$$Q(t) = p(t)\sin(2\pi f_D t) \quad (3.14)$$

hvor $p(t)$ beskriver pulsens omhyldningskurve. Pga at dopplermålingen gjøres gjennom et helt pulstog, vil også svært lave dopplerfrekvenser kunne bli målt ved denne metoden.

Figur 3.7 viser hvordan dette kan settes i system i en generell radar som utfører doppler prosessering. Dopplermålingene foretas ved å utføre diskret fouriertransformasjon (DFT) av samplene fra de enkelte avstandene gjennom et helt pulstog. Nærmere forklart foretas det en fouriertransformasjon for hver av radarens avstandsceller, se stiplet linje i figur som viser transformasjonen for en enkelt avstandscelle. Transformasjonen henter et sett I og Q-refleksjonssamplere fra hver av de N radarpulsutsendelsene med en gitt forsinkelse før transformasjonen starter. Figur 3.7 viser også hvordan I og Q-samplene kan organiseres i lagerlokasjoner for hver av de N radarpulsene og for hver av radarens avstandsceller.



Figur 3.7 Dopplerprosessering vha diskret fouriertransformasjon, ill. fra [6]

Hele prinsippet forutsetter at de utsendte radarpulsene er koherente, dvs at pulsene er utsnittet fra en kontinuerlig bærebølge.

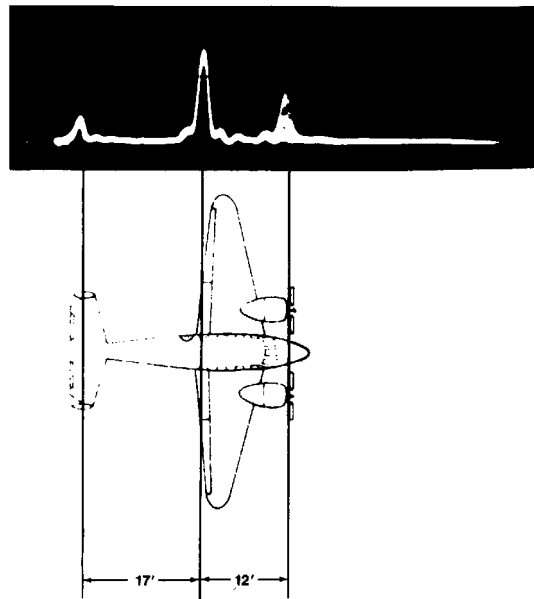
3.4 Eksempler på radarer som benytter koherent prosessering

3.4.1 High-Range-Resolution Radar (HRR)

HRR-radar, ref [7], er en type radar som har spesielt høy avstandsopløsning på grunn av at den utsendte bølgeformen inneholder pulskoding med høy kompresjonsgrad. Etter mottak av radarrefleksjonene kan radaren prosessere bølgeformen slik at høy avstandsopløsning oppnås. En HRR-radar kalles en avbildende radar selv om kun en avstandsprofil av radarmålet genereres. Denne profilen kan benyttes til identifikasjon og klassifikasjon av målet.

HRR-radarens utsendte bølgeform kan være kodet både med frekvens og fasemodulasjon. Typisk er bølgeformens kompresjonsgrad flere tusen ganger, og kan i enkelte tilfeller være så høy at avstandsopløsning under 1 meter oppnås. Ved generering av en avstandsprofil vil alle refleksjoner på de enkelte avstandene innenfor radarens antennelebe bli projisert ned på antennens siktelinje og vektorielt summert. Hver enkelt avstandscelle i radaren får dermed en amplitude og fase som er et mål for refleksjonens størrelse ved den enkelte radielle avstand.

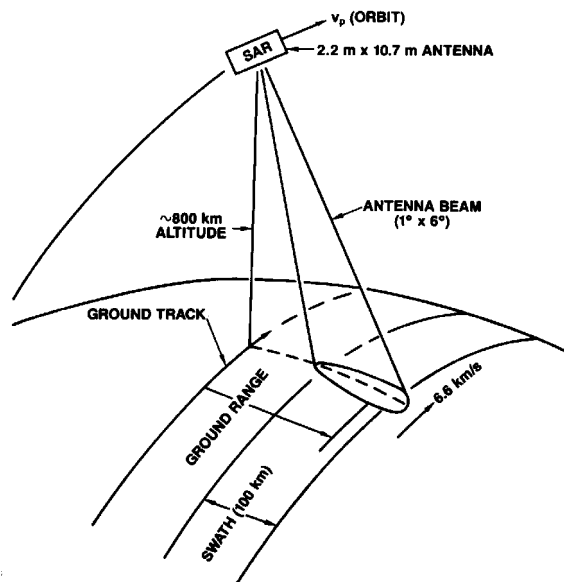
Figur 3.8 viser et eksempel på en avstandsprofil hvor enkelte karakteristiske parametere kan måles og som dermed kan benyttes til klassifisering av flytypen. Avstandsprofilene er høyst avhengige av flyets aspektvinkel og for et eventuelt automatisk gjenkjenningssystem må referanseprofiler for en rekke aspektvinkler innsamles.



Figur 3.8 Avstandsprofil med oppløsning 0.3 m av propellfly, ill. fra [7]

3.4.2 Synthetic Aperture Radar (SAR)

Ved å generere HRR-profiler over tid mens radarantennen forflyttes, og deretter foreta en dopplerprosessering eller SAR-prosessering av de innsamlede data, kan et SAR-bilde genereres, ref [7]. Flere forskjellige typer SAR-prosessering kan utføres avhengig av avstandsopløsning, diverse kompensasjoner eller fokuseringer eller operasjonelle ønskemål.



Figur 3.9 SAR-system, ill. fra [7]

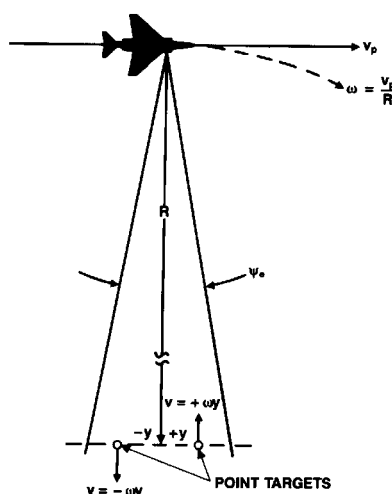
Et SAR-system kan ha følgende geometri som vist i Figur 3.9. Figuren viser en *Strip map SAR* hvor radaren avbilder striper av jordoverflaten fortløpende og som senere deles opp i

enkeltbilder. En annen mye brukt type er *Spotlight SAR* hvor bildet fokuseres spesielt på ett punkt på bakken under overflyvingen. Denne type SAR kan ha oppløsning mindre enn 1 meter, som er noe bedre enn hva *Strip map SAR* kan prestere i dag. Begge typer er *Side looking SAR* hvor radaren er montert på tvers av flyets eller satellittens fartsretning.

Radarens antenne har forholdsvis stort belyningsområde på bakken, slik at hver reflektor vil bli belyst en viss tid mens radaren passerer. Dermed har ikke radaren i utgangspunktet noen informasjon om hvor refleksjonen kommer fra, utenom tidsforsinkelsen og dermed avstanden fra radaren. For å bestemme reflektorens posisjon innenfor antennelobben må dopplerprosessering foretas.

SAR-bildets avstand langs radarens pekeretning kalles *slant-range* mens på tvers av pekeretningen kalles *cross-range*. For å måle en avstand i *slant-range* så måles tiden fra radarpulsen sendes ut til den mottas igjen etter refleksjon i terrenget. Dermed vil refleksjonen ligge på et sirkelsegment innenfor antennens belyningsvinkel.

Når det gjelder *cross-range* avstanden, så kan den finnes siden reflektorens dopplerskift utvikler seg gjennom scenens belysning. Som man kan se av Figur 3.10, vil først reflektoren ha positiv dopplerfrekvens så lenge reflektoren nærmer seg radaren, og deretter vil den bli negativ når den fjerner seg fra radaren. I det radaren peker rett mot reflektoren er dopplerskiftet lik null.



Figur 3.10 SAR cross-range Doppler, ill. fra [7]

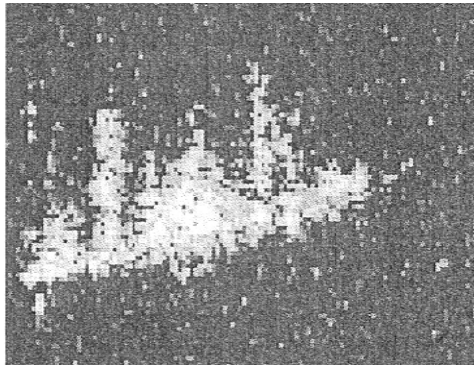
Ved å måle tiden fra radarpulsen sendes og til den mottas fra en refleksjon, samt å måle dens dopplerskift i mottakstidspunktet, kan refleksjonens posisjon bestemmes. Dette forutsetter imidlertid at reflektoren står stille. Ved bevegelige reflektorer vil denne måten å posisjonere dem på kunne feile, f.eks vil et tog i bevegelse kunne bli avbildet ved siden av togsportet!

Dagens SAR-radarer er hovedsaklig fokuserte SAR. Disse radarene har den egenskap at deres oppløsning i *cross-range* er uavhengig av *slant-range* på grunn av kompensasjon som gjøres under prosesseringen av bildet. Dermed fås et bilde med fast oppløsning i begge dimensjoner.

3.4.3 Inverse Synthetic Aperture Radar (ISAR)

I motsetning til en SAR-radar hvor prosesseringen baserer seg på at radaren forflytter seg i forhold til avbildende objekt, så utnytter ISAR-radaren, ref [7], objektets rotasjonsbevegelse for å avbilde objektet. Det er ellers mye felles signalbehandling i SAR og ISAR-avbildning.

ISAR-avbildning benyttes hovedsaklig for klassifisering og identifisering av luft-, land- og sjø-farkoster. Figur 3.11 viser et eksempel på ISAR-avbildning av et skip gjengitt i gråtoner, men i mange tilfeller blir refleksjonspunktene kodet i forskjellige farger avhengig av refleksjonens styrke.



Figur 3.11 ISAR-avbildning av et skip, ill fra [4]

4. Digitalt radio frekvens minne (DRFM) teknologi

4.1 Innledning

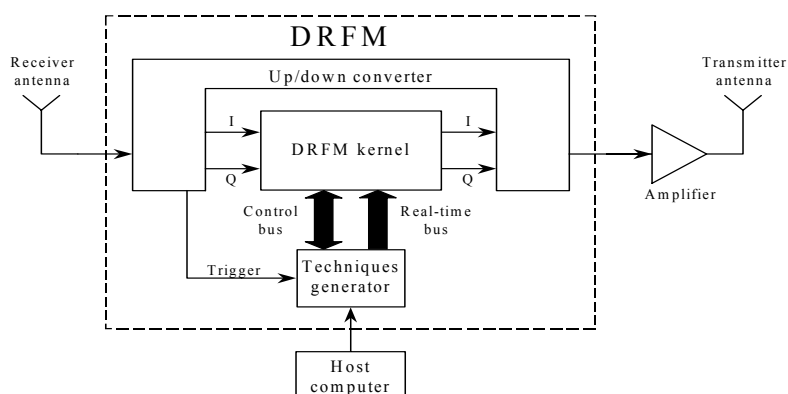
Jamming av radar foregår ved at man sender ut et signal for å forstyrre radaren. Dette jammesignalet må ha parametere som senterfrekvens, båndbredde, effekt osv som er slik at radaren påvirkes.

En type radarjammer søker å legge inn troverdige falske mål i radarbildet. Dette krever at både radarpulsene, og deres oppførsel og bevegelse er troverdige. Et utgangspunkt er her å bruke en kopi av radarens egne utsendte pulser, foreta manipulasjon i f.eks posisjon og deretter sende pulsen ut igjen. For at en slik manipulasjon skal kunne utføres, må en digital lagringsenhet og modulator for radarpulser finnes. Som nevnt i historiekapitlet så har dette blitt utviklet gjennom flere år og dagens teknologi tillater både store båndbredder og mange bit oppløsning.

Dette kapitlet baserer seg i stor grad på FFIs arbeider innen DRFM-utvikling, men med en god del egne betraktninger rundt temaet. Bakgrunnsinformasjon er hentet fra kildene [2] - [4], [14] - [15] og [25] - [30].

4.2 Blokkskjematisk beskrivelse

Figur 4.1 viser en generell oppkobling av en DRFM med effektforsterker og antenner. DRFM-enheten blir i figuren styrt fra en kontrollerende datamaskin, og i mange tilfeller vil dette være et intelligent mottakersystem som både gjør datainnsamling og analyse av aktuelle radarer.



Figur 4.1 Blokkskjema for en generell DRFM-oppkobling

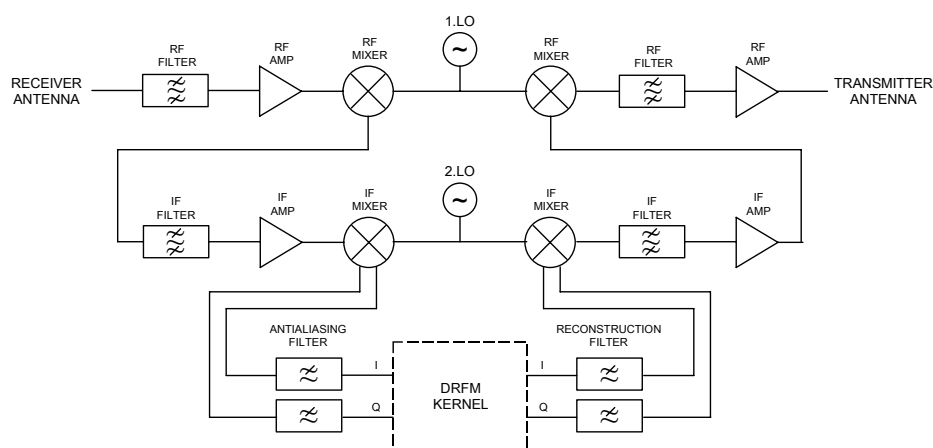
Selve DRFM-enheten består av 3 hoveddeler; mikrobølge opp/ned-konverter, DRFM-kjerne og teknikkgenerator.

4.3 Mikrobølge opp/ned-konverterer

For at radarsignalene i GHz-området skal kunne samples og digitaliseres må de frekvenskonverteres ned til basebånd i MHz-området. Den samme prosessen må reverseres for å bringe jammesignalene tilbake til GHz-området etter lagring og manipulasjon i DRFM-kjernen.

Figur 4.2 viser et blokkskjema for en generell opp/ned-konverterer med 2-steps frekvenskonvertering. Fra mottakerantennen blir radarsignalet båndfiltrert og forsterket før miksing med første lokaloscillator. Utgangssignalet $IF = RF - 1.LO$ føres videre til neste nivå. Mellom miksertrinnene er det filtrering og forsterkning for å fjerne uønskede signalkomponenter eller spuriøse signaler, og for å gjenvinne signalnivået. En eller flere av forsterkerne i opp/ned-konvertereren er ofte av en type som begrenser utgangssignalets dynamikk, dvs har tilnærmet fast utgangsnivå. Mellomfrekvensmikseren er en kvadraturmikser som leverer ut basebånd I og Q-signaler, $BB = IF - 2.LO$. Disse signalene blir så filtrert for å fjerne signalkomponenter høyere enn Nyquistfrekvensen før overføring til DRFM-kjernen.

Frekvenskonverteringen fra basebånd og tilbake til RF-frekvensene utføres på samme måte, bare i motsatt rekkefølge.



Figur 4.2 Blokkskjema for mikrobølge opp/ned-konverter

Et viktig poeng er at det brukes de samme lokaloscillatorene, ref Figur 4.2 – 1.LO og 2.LO, til både opp og ned-konvertering. Dette sørger for at radarsignalet inn og jammesignalet ut er fasekoherent, noe som er helt essensielt hvis et falskt radarmål skal bli troverdig på f.eks en HRR-radar.

Et annet viktig poeng er at siden manipulasjonen i DRFM-kjernen gjøres i basebånd, så må det huskes på at det ikke er samsvar mellom en forsinkelses fasedreining av signalet i RF-båndet og i basebåndet. Tiden en forsinkelse varer vil bli lik, men den fasedreiningen signalet får under den samme forsinkelsen er gitt av basebåndfrekvensen i stedet for RF-frekvensen. Selv om signalet i basebånd frekvenskonverteres tilbake til RF-båndet etter forsinkelsen, vil det likevel ikke være samsvar mellom aktuell forsinkelse og dens fasedreining av signalet. Ligning (4.1) – (4.8) viser dette nærmere:

$$\text{Lokaloscillator :} \quad \omega_{LO} = \omega_{1.LO} - \omega_{2.LO} \quad (4.1)$$

$$\text{Basebånd :} \quad \omega_{BB} = \omega_{RF} - \omega_{LO} \quad (4.2)$$

Jammeren mottar radarsignalet:

$$\text{Jammer Rx :} \quad e^{j\omega_{RF}t} \quad (4.3)$$

Jammeren mikser radarsignalet ned til basebånd:

$$\begin{aligned} \text{Jammer basebånd}_{inn} : \quad & e^{j\omega_{RF}t} \cdot e^{-j\omega_{LO}t} \\ & = e^{j(\omega_{RF} - \omega_{LO})t} \end{aligned} \quad (4.4)$$

Jammeren forsinker radarsignalet et tidsrom τ :

$$\text{Jammer basebånd}_{ut} : \quad e^{j(\omega_{RF} - \omega_{LO})(t - \tau)} \quad (4.5)$$

Jammeren mikser det forsinkede radarsignalet tilbake til RF:

$$\begin{aligned} \text{Jammer Tx :} \quad & e^{j(\omega_{RF} - \omega_{LO})(t - \tau)} \cdot e^{j\omega_{LO}t} \\ & = e^{j[\omega_{RF}(t - \tau) + \omega_{LO}\tau]} \end{aligned} \quad (4.6)$$

Når jammesignalet mottas i radaren blir det mikset med kompleks konjugerte av utsendt radarsignal:

$$\begin{aligned} \text{Radar}_{signalbehandling} : \quad & \text{Radarsignalet}^* \cdot \text{Jammer Tx} \\ & = e^{-j\omega_{RF}t} \cdot e^{j[\omega_{RF}(t - \tau) + \omega_{LO}\tau]} \\ & = e^{-j(\omega_{RF} - \omega_{LO})\tau} \\ & = e^{-j\omega_{BB}\tau} \end{aligned} \quad (4.7)$$

Ligning (4.7) viser at jammesignalet har fått feil fasedreining etter frekvenskonvertering til basebånd, forsinkelse og så konvertering tilbake. Fasedreiningen er her gitt av ω_{BB} og τ . Til sammenligning viser ligning (4.8) uttrykket for radarbåndet for en ideell refleksjon. Her er fasedreiningen gitt av ω_{RF} og τ .

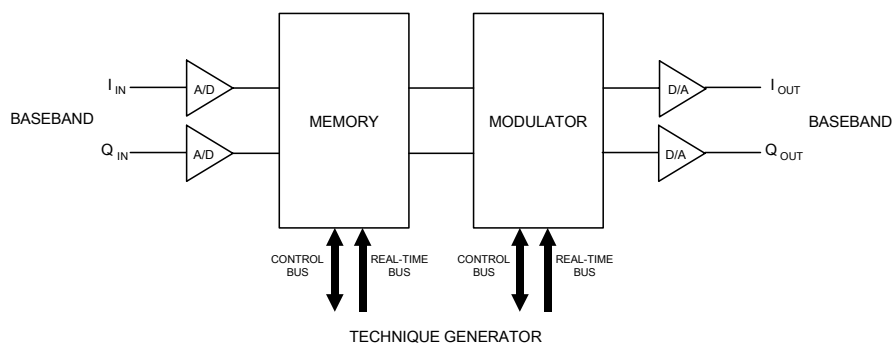
$$\begin{aligned} \text{Radar}_{signalbehandling} : \quad & \text{Radarsignalet}^* \cdot \text{Refleksjon} \\ & = e^{-j\omega_{RF}t} \cdot e^{j\omega_{RF}(t - \tau)} \\ & = e^{-j\omega_{RF}\tau} \end{aligned} \quad (4.8)$$

Dermed må en forsinkelse utført i basebånd korrigeres med en fasedreining av signalet.

4.4 DRFM-kjerne

DRFM-kjernen sørger for konvertering av de analoge I og Q-signalene til digitalt format i A/D-konverterne. Deretter lagres de digitale samplene i RAM-hukommelse i påvente av utlesing. Kjernen vil først på kommando fra teknikkgeneratoren starte opp utlesing fra RAM-hukommelsen igjen og vil da sende datastrømmen gjennom modulatorene på vei til D/A-konverterne. Modulatoren genererer komplekse overlappende radarmål av de enkle radarpulsene som mottas, og tilslutt i D/A-konverterne blir de analoge I og Q-signalene gjenskapt før overføring til mikrobølge opp/ned-konverter.

Figur 4.3 viser et blokkskjema for DRFM-kjernen med dens hoveddeler.



Figur 4.3 Blokkskjema for DRFM-kjerne

4.4.1 Analog-til-digital konverterne (A/D)

Kravene til A/D-konverterne er høye på grunn av den svært høye samplingshastigheten. Tidligere hadde A/D-konvertere med høy hastighet kun ett eller noen få bit oppløsning, men i dag finnes A/D-konvertere med 500 MHz samplerate og 12 bit oppløsning i amplitude, ref [25], og 1,5 GHz samplerate og 8 bit oppløsning i amplitude, ref [26].

A/D-konverternes samplerate velges ut i fra høyeste signalfrekvens i inngangssignalet, da sampleringen må være minst 2 ganger høyeste signalfrekvens for å overholde Nyquistteoremet. Avhengig av om det benyttes kvadratur sampling eller ikke, vil båndbreddens verdi maksimalt bli lik sampleringen eller det halve. I praksis vil båndbredden bli noe mindre pga antialiasingfiltrene som skal hindre signalkomponenter over Nyquistfrekvensen fra å bli tilført A/D-konverternes innganger.

Fordelen med kvadratur sampling er den doble båndbredden, men dette fører til mer maskinvare i form av doble signalveier helt fra mikrobølge ned-konverterens utganger, gjennom A/D-konvertering, RAM-hukommelse, modulator, D/A-konvertering og til opp-konverterens innganger. Dagens komponenter har gode toleranser slik at problem med ubalanse mellom I og Q-kanalene ikke er tilstede, slik det ofte var tidligere.

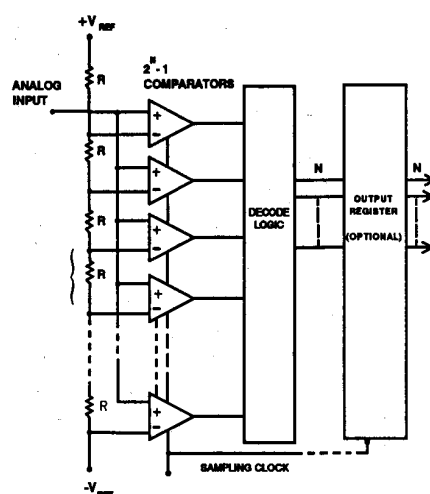
Jo høyere oppløsningen som A/D-konverterne kan digitalisere de analoge signalene med, jo bedre vil signal-til-støyforholdet bli. Vanligvis vil også mikrobølge opp/ned-konverter ha automatisk nivåkontroll, slik at A/D-konverternes inngangsnivå blir optimalisert og hele

konverterens dynamikk blir utnyttet til enhver tid. I tillegg må det legges merke til at jammeren mottar et direktesignal fra radaren og den trenger dermed ikke å forsterke opp en svak radarrefleksjon fra bakgrunnsstøyen. Alt i alt vil en relativ beskjeden oppløsning i amplitude være tilstrekkelig, som f.eks i EKKO II jammeren hvor det benyttes amplitudeoppløsning på 4 bit i I-kanal og 4 bit i Q-kanal, ref [2].

A/D-konvertere flest digitaliserer inngangssignalets amplitude til ett av flere diskrete nivåer, men enkelte spesielle konvertere digitaliserer inngangssignalets fase i stedet. Representasjon av et gitt signal med fasekoding vil være mer kompakt når det gjelder lagringsplass enn et tilsvarende signal med amplitudekoding pga at et fasekodet signal pr definisjon har amplitude lik 1 – amplitudeinformasjonen trenger dermed ikke lagres. Viktigere er det at et fasekodet signal vil kunne gjenskape konstant amplitude i et system med varierende demping over frekvensområdet.

Digitaliseringsstøyen i et amplitudedigitalisert system er 'hvit' på grunn av støyens tilfeldige natur, ref [11], mens i et fasedigitalisert system vil den være harmonisk pga at signalet diskretiseres til faste fasevinkler. Enkelte DRFM-systemer kan benytte en kombinasjon av amplitude- og fasekoding, pga at A/D-konvertere med amplitudekoding er kommersielt tilgjengelig, mens fasekodingens fortrinn ved fasemodulasjon ønskes utnyttet.

A/D-konvertere kan konstrueres i flere forskjellige arkitekturer. Den enerådende typen for høyhastighetskonvertere er 'flash' arkitekturen, ref [25], som har en parallell oppbygning. Figur 4.4 viser en skisse av denne arkitekturen.



Figur 4.4 Flash arkitektur for høyhastighets A/D-konvertere, ill. fra [25]

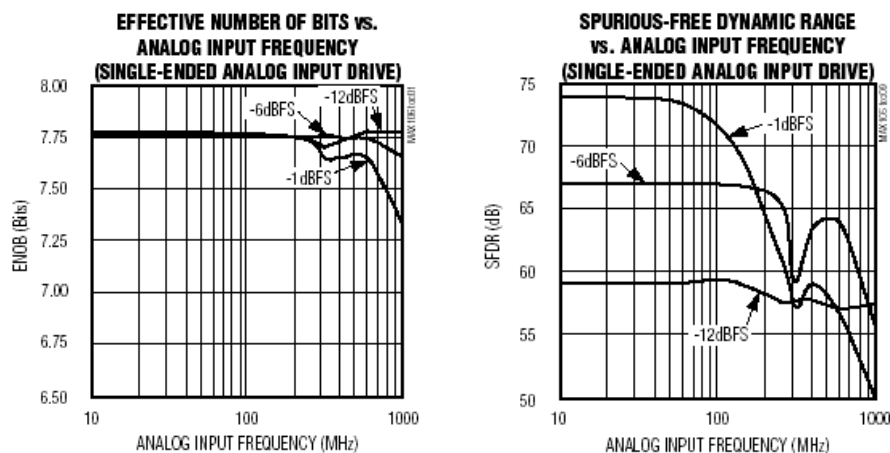
Flash-konvertere er bygget opp med $2^n - 1$ spenningskomparatorer, hvor n er antall digitaliseringsnivåer. Spenningskomparatorene sammenligner inngangssignalet med hver sin avtapning i et tilhørende forspent motstandsnettverk. Jo høyere innspenning, jo flere komparatorer snus om. Alle komparatorene tilkobles deretter logikk for å omsette deres tilstander til en binærkode. Ved å bruke mange komparatorer oppnås mange digitaliseringsnivåer og mange bit oppløsning.

Et forhold som krever oppmerksomhet er konverterens antall effektive bit. I forskjellige datablad er vanligvis en kurve med antall effektive bit versus signalfrekvens tegnet, se Figur

4.5 venstre side. Kurven forteller hva beregnet antall bit blir om det målte signal-til-støyforholdet ($SNR_{M\grave{A}LT}$) legges til grunn i motsetning til det teoretiske signal-til-støyforholdet gitt av digitaliseringsstøyen. Ligning (4.9) viser utregning av antall effektive bit, ref [25].

$$\text{Antall effektive bit} = \frac{SNR_{M\grave{A}LT} - 1.76dB}{6.02} \quad (4.9)$$

En annen kurve som vanligvis også presenteres er spuriøs fri dynamisk område versus signalfrekvens, se Figur 4.5 høyre side. Denne kurven viser differansen mellom nyttesignalet og den høyeste spuriøse komponenten ved forskjellige signalfrekvenser. Begge kurvesettene blir vanligvis målt ved -1dBFS, -6dBFS og -12dBFS (dBFS – referert til full skala).



Figur 4.5 Kurver for effektivt antall bit og spuriøs fri dynamisk område, ill. fra [26]

4.4.2 Datahukommelse

Ved høye samplerater blir datahukommelsen nødt til å motta både hurtige og store mengder data. Hukommelsen sett under ett, må kunne håndtere syklustider i området 1-5 ns gitt aktuell samplingshastighet. Dette håndteres med tidsmultipleksing i forskjellige kanaler og deretter sammenstilling av datasamplene til store ordlengder. Da kan flere høyhastighets RAM-kretser med syklustid på rundt 10 ns benyttes og settes sammen til store bitbredder.

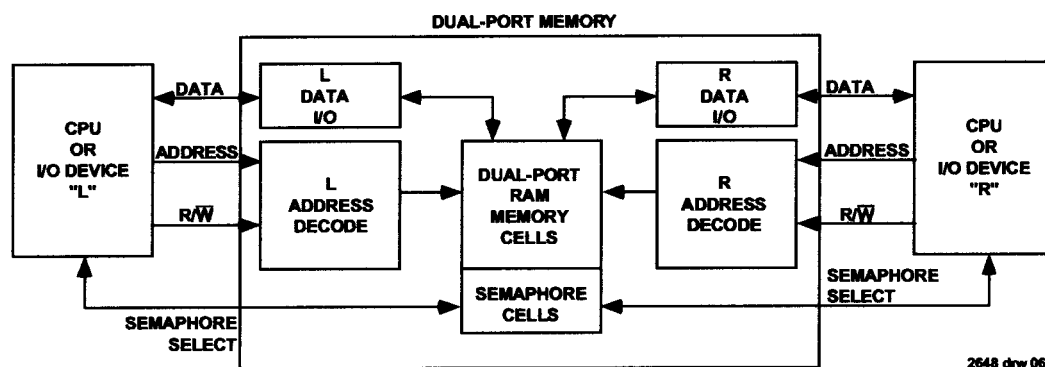
Som nevnt gir høye samplerater store mengder data. Hukommelsens størrelse må dimensjoneres etter maksimal lengde på forventet radarpuls, antall bit digitalisering og på antall radarpulser som kan tenkes å lagres samtidig. En enkelt radarpuls kan ha en varighet på opp mot 250 us og med en samplerate på 1 GSPS blir lagringsbehovet 250000 lagerlokasjoner. Ved bruk av kvadratur sampling og 6 bit oppløsning i både I og Q må hver lagerlokasjon ha plass til 12 bit med data.

Tilslutt i regnestykket må det tas med at det ofte er ønskelig å kunne lagre flere radarpulser samtidig, pga at i enkelte situasjoner ofte vil være flere aktive radarer og et minimum må være å lagre en puls fra hver av dem før hukommelsen overskrives av neste puls. Et typisk antall samtidige lagrede radarpulser kan være 4.

Dette eksemplet fører til følgende regnestykke:

250000 samples x 12 bit x 4 radarpulser = 12 Mbit

For at hukommelsen skal kunne være i stand til å både lagre radarpulser og lese ut pulser for generering av jammesignal samtidig, må hukommelsen være av 2-port type. Denne typen hukommelse, Figur 4.6, har to helt separate porter med adresse, data og kontrollsignaler, og kan opereres helt samtidig og uavhengig på de to portene.



Figur 4.6 2-port hukommelse for samtidig tilgang fra to enheter, ill. fra [27]

I tillegg til selve lagerlokasjonene trengs det en viss styring og kontroll av hukommelsen. Til dette kan programmerbar logikk benyttes og spesielt er FPGA-kretsene anvendelige. Fordelene med bruk av FPGA er store i forhold til diskrete komponenter; rekonfigurerbare, stor logikk-kapasitet og intern hukommelse på lite areal, og støtter mange forskjellige I/O-standarder blant annet.

Datahukommelsen har behov for å holde oversikt over hvor i hukommelsen de forskjellige radarpulsene er lagret, hvor skal neste mottatte puls lagres osv. Hukommelsen blir ofte delt opp i flere områder eller trusselfiler, en for hver av radartruslene som detekteres. Dette fordrer at det finnes logikk som er i stand til å gjøre en rask analyse av pulssamplene og kunne bestemme enkelte pulsparametere som pulsens frekvens i starten av pulsen, frekvens i slutten av pulsen, pulsens varighet osv. Når de mottatte pulsene kan gjenkjennes på denne måten kan pulsene fra en radar tilordnes en fil, og pulsene fra en annen radar en annen fil.

Dermed blir fleksibiliteten stor i forhold til hvilke trusler som skal jammes og hvilke som ikke skal. Dette er det derimot teknikkgeneratoren som håndterer, ref kapittel 4.5.

4.4.3 Modulator

DRFM-kjernens modulator har som oppgave å generere et eller flere falske komplekse radarmål for hver radarpuls som leses ut fra datahukommelsen. Med dagens direkte realiserte modellering vil hver av de komplekse målene bygges opp av enkeltreflektorer hvor forsinkelse i tid og modulasjon i fase og amplitude kan utføres. Teknikkgeneratoren som er beskrevet i kapittel 4.5, sørger for å overføre modulasjonsparametere etter hvert som modulasjonen skal utføres.

Tidsforsinkelsene er programmerbare i steg gitt av sampleraten i systemet og dekker de enkelte reflektorenes forskjell i avstand innen et komplekst radarmål. Dette kan realiseres ved hjelp av FIFO-hukommelse som mellomlagerer radarpulsen og leser den ut igjen etter en

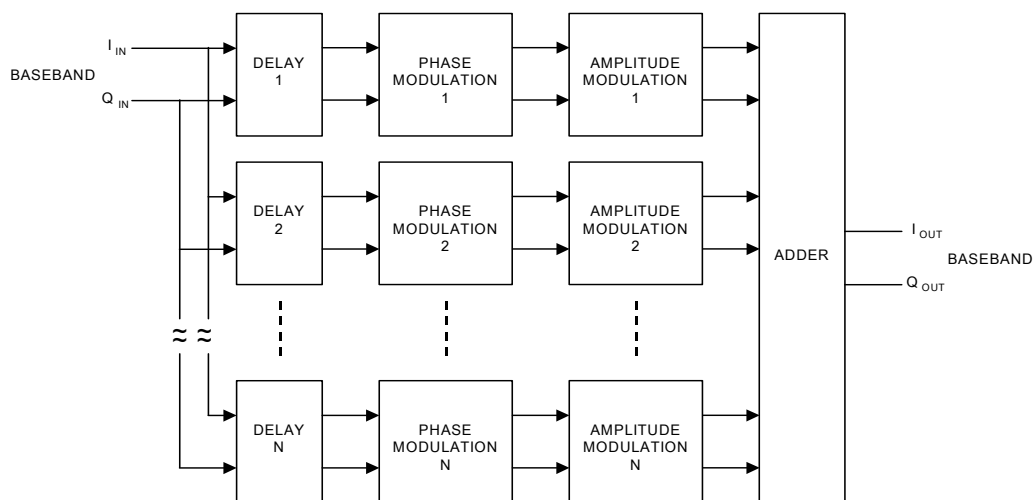
gitt tid. Den ekstra tidsforsinkelsen som FIFO-hukommelsen utgjør, vil få radaren til å gjengi reflektoren med litt større avstand enn de andre reflektorene i det komplekse radarmålet.

Fasemodulasjon av en refleksor, som endres over tid, vil generere et dopplerskift som gjør at radaren oppfatter reflektorens hastighet som en annen enn jammerens hastighet.

Amplitudemodulasjon kan statisk benyttes for å fastsette reflektorenes innbyrdes amplitude, men kan også benyttes aktivt sammen med fasemodulasjonen til å generere flere samtidige dopplerfrekvenser.

Til slutt skal alle reflektorene summeres vektorielt sammen til et resulterende jammesignal. Jammesignalet blir dermed en sum av mange fase- og amplitudemodulerte overlappende kopier av det opprinnelige radarsignalet.

Figur 4.7 viser skjematisk hvordan dette kan gjøres. Hver av grenene kan i utgangspunktet bare generere en enkelt refleksor med en gitt avstand og hastighet. Ved aktiv bruk av både fase og amplitudemodulasjon samtidig, kan flere dopplerfrekvenser genereres og flere reflektorer med samme avstand, men med forskjellige hastigheter oppstår.



Figur 4.7 Skjematisk oppbygning av modulator

Som nevnt under *Datahukommelse*, avsnitt 4.4.2, så er FPGA-kretser anvendelige i mange sammenhenger. Spesielt gjelder dette når store mengder logikk inneholdende både aritmetiske beregninger og mindre hukommelseskretser skal implementeres. Noen familier av FPGA-kretsene er spesialisert til å håndtere slike problem, med mengder av innebygde 'harde' multiplikatorer og hukommelsesområder i tillegg til vanlig logikk.

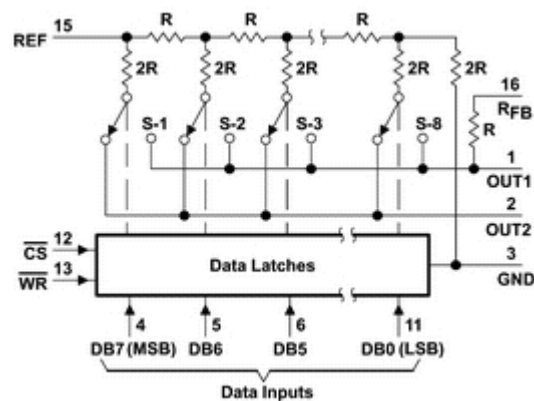
Ved hjelp av disse spesialiserte FPGA-kretsene kan konvertering mellom amplitudekodede og fasekodede signaler utføres effektivt ved oppslagstabeller. Dermed kan fasemodulasjon utføres i frekvensdomenet ved hjelp av addisjon og amplitudemodulasjon gjøres i tidsdomenet med multiplikasjon.

4.4.4 Digital-til-analog konverterne (D/A)

D/A-konverterne gjensker analoge signaler fra I og Q-kanal datastrømmene. På grunn av at jammesignalet får en langt større kompleksitet enn mottatt radarpuls, så er behovet for flere bit amplitudeoppløsning tilstede. En tommelfingerregel er at for hver gang antall samtidige reflektorer dobles, så må jammesignalets bitoppløsning øke med ett bit.

Siden de fleste DRFM-jammere ikke er i stand til å generere komplekse overlappende radarmål, så benyttes det like stor amplitudeoppløsning i A/D og D/A-konverterne. Men, EKKO II jammeren har denne egenskapen og har 4 bit oppløsning i A/D-konverterne og 10 bit oppløsning for bruk til komplekse mål i D/A-konverterne.

En D/A-konverter er i prinsippet bygget opp ved hjelp av et sett strømvendere, et R-2R motstandsnettverk og en differensiell strømførsterker. Strømvenderne, styrt av hver sin datainnngang, kobler de enkelte vektene gitt av R-2R nettverket til henholdsvis positiv eller negativ strømutgang. Figur 4.8 viser en generell D/A-konverter arkitektur.



Figur 4.8 Generell D/A-konverter arkitektur, ill. fra [30]

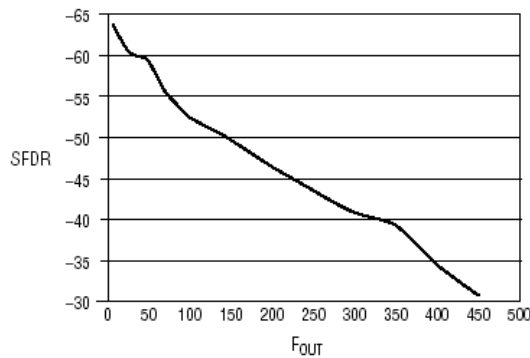
Resultatsignalet er differansen mellom strømmene OUT1 og OUT2, og må forsterkes i en tilhørende differensiell strømførsterker som ofte ikke er en del av selve D/A-konverterkretsen. Den er heller ikke med på Figur 4.8.

Ulike parametere benyttes ved spesifisering av D/A-konvertere, ref [25] og [26]. Spesielt er parameteren *settling time* viktig ved høyfrekvenskonvertere. Den angir hvor fort den analoge utgangen stabiliserer seg etter påtrykk av nye data og er derfor avgjørende for hvor høyfrekvent utgangssignalet kan være.

En annen vanlig brukt parameter er spuriøsfri dynamisk område som angir avstanden mellom nyttesignalet rms-verdi og høyeste spuriøse rms-verdi. I signal-til-støyforhold beregninger inngår den totale støyen som er rms-summen av alle spuriøse over båndbredden opp til Nyquistfrekvensen og derfor blir signal-til-støyforholdet dårligere enn det spuriøsfrie dynamiske området.

Det spuriøsfrie dynamiske området er sterkt avhengig av samplingsfrekvensen og det aktuelle utgangssignalets frekvens. Figur 4.9 er et eksempel på en 1 GSPS 14 bit D/A-konverter med oppgitt spuriøsfri dynamisk område på 52 dB ved 75 MHz utgangssignal. Med et utgangssignal på 400 MHz har det spuriøsfrie området sunket til 34 dB. Selv om

denne D/A-konverteren har høy samplerate kan den bare benyttes til relativt lavfrekvente utgangssignaler uten at de spuriøse komponentene øker kraftig.



Figur 4.9 Spuriøsfri dynamisk område versus utgangssignalets frekvens på en 1000 MSPS D/A-konverter med 14 bit oppløsning, ill. fra [29]

D/A-konvertere kan i dag skaffes med oppdaterings hastighet på 1 GSPS og med oppløsning 14 bit, ref [29]. Endog 16 bit konvertere er tilgjengelige, riktignok bare med hastighet på 600 MSPS, ref [25].

4.5 Teknikkgenerator

Teknikkgeneratoren er den delen av DRFM-systemet som sørger for å beregne modulasjonsdataene som DRFM-kjernen benytter under manipulasjonen. Beregningene som skal gjøres kan grovt deles i to; de som er tidskritiske og som må oppdateres hurtig, og de som ikke er det.

I den første gruppen tilhører beregning av dopplersignalenes fasevektorer. Fasene må jevnlig oppdateres slik at dopplerfrekvensene utvikles med tiden. Hvis multiple dopplerfrekvenser skal genereres i de enkelte grenene i modulatorene, ref Figur 4.7, blir faseberegningene flere og amplitudemodulasjonen må beregnes og oppdateres samtidig.

Alle disse beregningene må enten utføres ved hjelp av en dedikert signalprosessor eller i spesialutviklet maskinvare. Den mest fleksible løsningen er nok FPGA i dag. Noen av de største kretsene, ref [28], tillater bruk av et stort antall (~500) dedikerte multiplikatorer for signalbehandlingsformål. Dette gjør at et stort antall beregninger kan gjøres i parallell med hurtig logikk, og alt kan plasseres i en enkelt FPGA-krets.

De beregningene som ikke er tidskritiske, som beregning av reflektorers forflytning over tid og andre generelle kontrollfunksjoner, kan utføres ved hjelp av prosessortechnologi i et standard prosessorsystem. Men et slikt system vil ofte kreve en del komponenter og oppta et visst volum. En annen løsning kan være et *'embedded'* prosessorsystem hvor enten en prosessorkjerne i silisium i en FPGA tas i bruk ('hard' prosessor) eller det legges inn en eller flere *'myke'* prosessorer i en standard FPGA. 'Myke' prosessorer er realisert i VHDL-kode og kan legges inn i de fleste typer FPGA-kretser. Til større sanntids kontrolloppgaver kan FPGA-løsninger med kombinasjoner av dedikert parallell signalbehandlingslogikk, en 'hard' kraftig prosessor og flere 'myke' mindre prosessorer være en god løsning.

Som beskrevet ovenfor kan både tidskritiske og ikke-tidskritiske beregninger og rene kontrollfunksjoner samles i en felles FPGA-krets i dag. Løsningen er plassbesparende, effektbesparende og med høy ytelse.

4.6 EKKO II – Eksperimentell DRFM radarjammer

EKKO II, ref [2], er bygget opp rundt et rammeverk av hyllevarer 19" rackbokser. Figur 4.10 viser nederst – Digital Unit, inneholder DRFM-kjernen med integrert teknikkgenerator, bestående av høyhastighets A/D- og D/A-konvertere, hukommelse og digital signalprosessering/modulator kretskort. Den midterste boksen – RF Unit, består av mikrobølge 'front-end', som konverterer radarsignalene opp og ned til basebånd. Den øverste boksen er en fabrikkprodusert effektforsterker for radarfrekvenser. I tillegg benyttes en standard PC-skjerm og tastatur, samt to antenner for mottak og sending av radarsignalene.

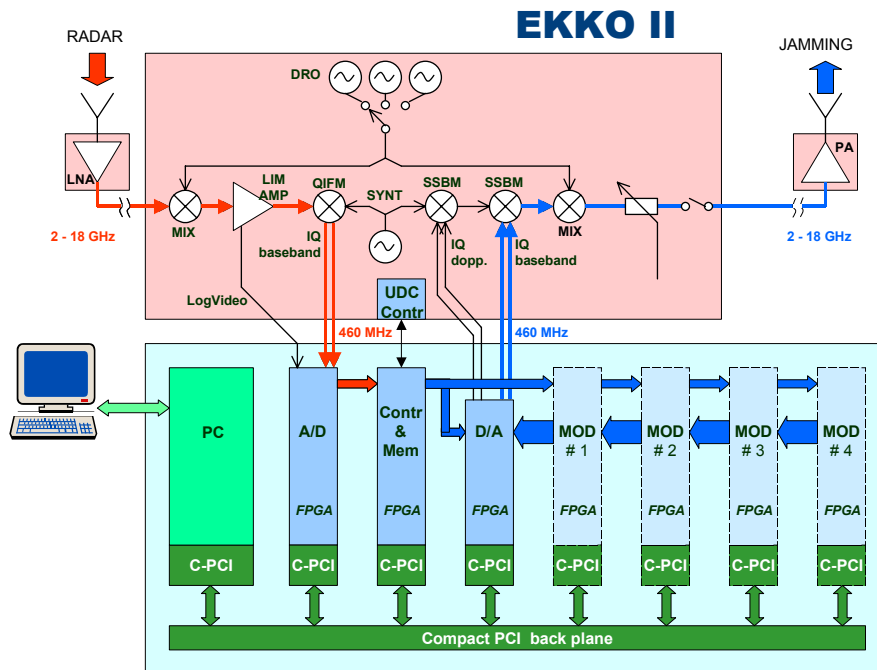


Figur 4.10 EKKO II DRFM-basert radarjammesystem, ill. fra [2]

Arkitekturen i EKKO II-systemet er illustrert i Figur 4.11. DRFM-enheten blir på høyt nivå kontrollert av en standard 'single-board' PC via *Compact PCI*-buss (cPCI). DRFM-kjernen består ellers av fire forskjellige typer kretskort (A/D, D/A, hukommelse og modulator) som kommuniserer via dedikerte høyhastighets dataforbindelser i et bakplan. Det er avsatt fire kretskortposisjoner for modulorkortene som vil få en skalerbar oppbygning, slik at systemet kan bestykes med kort etter behov. Alle kretskortene i DRFM-kjernen benytter kommersielt tilgjengelige FPGA-kretser til både kontrollfunksjoner og til den digitale signalprosesseringen. FPGA-kretsene bidrar til en modulær og reprogrammerbar arkitektur.

Signalflyten gjennom systemet er som følger, ref Figur 4.11: Radarsignalet fanges opp av mottakerantennen og føres via 'RF front-end' til A/D-kortet, og deretter til kontroll/hukommelsekortet for lagring (rød signalvei). Ved ønsket utsendelsestidspunkt føres en pulskopi (blå signalvei) enten direkte til D/A-kortet eller via ett eller flere

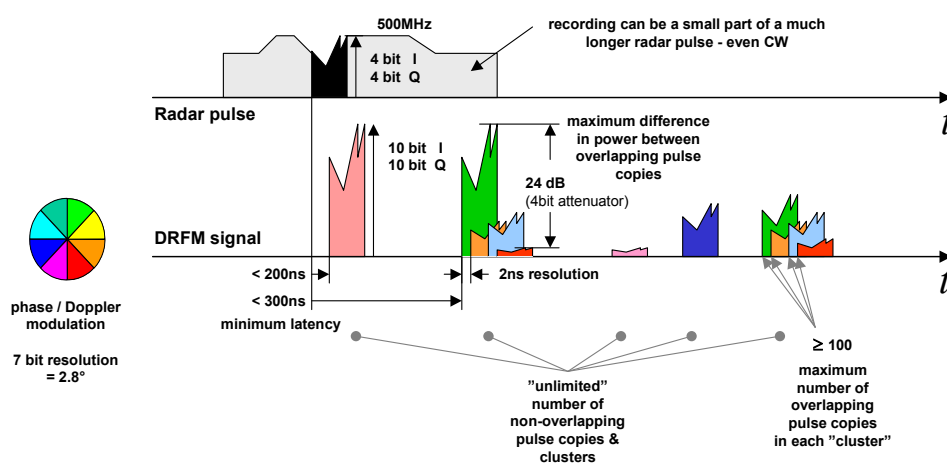
modulorkort, og deretter sendes jammesignalet til RF-delen, effektforsterker og til slutt senderantennen.



Figur 4.11 Arkitekturen i EKKO II-systemet, ill. fra [2]

Hvert enkelt mottatt radarsignal kan individuelt forsinkes, fase- og amplitudemoduleres før gjenutsending. De fire identiske modulorkortene vil kunne modellere troverdige radarmål med utstrekning mot høyoppløselige radarer ved hjelp av et antall overlappende pulskopier, hver enkelt med individuell amplitude og fase. Modulorkortenes prosesseringsstruktur kan endres under drift ved nedlasting av ønsket FPGA-programkode.

Figur 4.12 viser en illustrasjon over hvilke muligheter som ligger inne i EKKO II-systemet:



Figur 4.12 Forskjellige muligheter for pulsmodulasjon i EKKO II

Figur 4.12 viser at hele eller vilkårlige deler av et radarsignal kan brukes som grunnlag for modellering av jammesignal i EKKO II. Jammesignalet eller DRFM-signalet i Figur 4.12 er fargelagt for å indikere individuell fasemodulasjon på de enkelte overlappende pulskopiene.

Pulskopiene kan posisjoneres med 2 ns oppløsning i forsinkelse som tilsvarer 0,3 m i avstand. Det er lagt opp til en dynamikk på maksimalt 24 dB mellom de enkelte pulskopiene i jammesignalet.

Tabell 4.1 lister opp EKKO IIs karakteristiske parametere i kortform:

Frevensområde	2-18 GHz
Instantan båndbredde	460 MHz (-3dB)
A/D-konverter	4bit I + 4bit Q kvadratur / 500 MHz sampling
D/A-konverter	10bit I + 10bit Q kvadratur / 500 MHz sampling
Antall enkle falske mål	> 1000 (Koordinert i avstand og hastighet)
Antall komplekse falske mål	> 100 (Overlappende pulser med individuell kontroll av doppler og amplitude)
Antall trusselfiler	~8 programmerbart
Hukommelseskapasitet	2 Mbit = 362k sample = 524 us
Forsinkelsesoppløsning	2 ns
Mottakerfølsomhet	-60 dBm – 0 dBm programmerbart
Sanntid pulsparameter målinger	Ankomsttid, frekvens, amplitude, pulsbredde, pulsrepetisjonsintervall (PRI), puls rådata

Tabell 4.1 Spesifikasjoner på EKKO II eksperimentell DRFM radarjammer

5. Modelling av radarrefleksjoner

5.1 Innledning

Dette kapitlet tar spesielt for seg modellering av komplekse refleksjoner brukt mot HRR-radar. Resultatet av jammingen skal fremstå som troverdige avstandsprofiler etter pulskompresjon i radaren. Arbeidene viser sammenhengen mellom utsendt bølgeform, dens kompresjonsgrad og sidelobenivå, og dens evne til å løse opp de enkelte reflektorene i en kompleks radarrefleksjon.

Årsaken til at dette kapitlet i så stor grad fokuserer på HRR-avstandsprofiler, er at disse avstandsprofilene danner grunnlaget for all modellering av komplekse refleksjoner i alle avbildende radarer. De mer avanserte SAR og ISAR-radarene vil også kunne modelleres med realiseringskonseptene som utvikles for HRR-radar, men DRFM-modulatoren vil da bli tilført et enda mer komplekst modulasjonssignal som endres hurtig fra en utsending til en annen.

Første del av kapitlet ser generelt på aktuelle metoder for modellering. Dagens metode med direkte realisert modellering er utgangspunktet, og deretter blir forslaget med beregnet realisert modellering gjennomgått, ref [1]. Dette forslagens belastninger i antall komplekse regneoperasjoner blir sammenlignet ved modellering i både tids- og frekvensdomenet.

Siste del av kapitlet omhandler datamodellering for verifisering av forskjellige metoder og realiseringskonsepter som er blitt utviklet under hovedfagsarbeidet. Programverktøyet MatLab er blitt benyttet for å lage en datamodell av de forskjellige DRFM-modulatorkonseptene, samt modeller for radarens utsendte signaler og radarens pulskompresjon etter mottak av jammesignalene.

5.2 Aktuelle metoder for radarjammer modellering

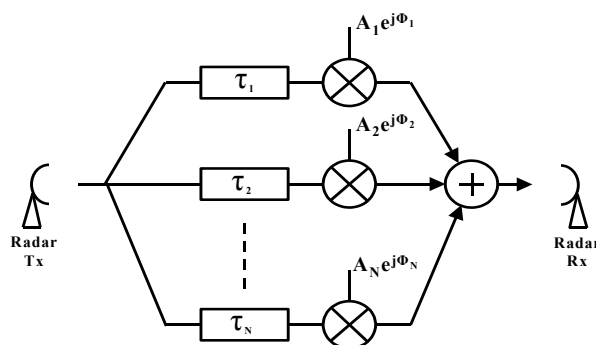
I utgangspunktet skal modelleringen tilsvare den fysiske prosessen som utspiller seg i den virkelige verden. Der vil hvert enkelt refleksjonspunkt hos et gitt objekt sørge for refleksjon med gitt forsinkelse, amplitude og fase. Radarens bølgeform som belyser objektet vil i mange tilfeller ha større varighet i tid enn refleksjonspunktene innbyrdes avstand på objektet, gitt radarsignalet forplantningshastighet. Da refleksjonene ligger tett i avstand og tid, vil resultatsignalet til enhver tid bestå av vektorsummen av alle de samtidige refleksjonene. Reflektorenes innbyrdes faseforskjell vil besørge interferens slik at noen reflektorer forsterkes og andre svekkes. I en radar vil avstandsområdet bli delt inn i et antall avstandsceller. De enkelte cellene vil ha en amplitude og fase som er et mål for summen av alle reflektorenes amplituder og faser i den gitte celle.

I en tradisjonell radar vil den utsendte bølgeformen ha så lang varighet i tid at et objekts forskjellige reflektorer vil bli belyst samtidig og blir dermed representert som ett mål i

radarens avstandsceller. I en HRR-radar derimot, ref avsnitt 3.4.1, er bølgeformen etter pulskompresjon så kort at de enkelte reflektorene skilles i forskjellige avstandsceller, gitt aktuell avstandsopløsning. På grunnlag av informasjonen i radarens avstandsceller kan avstandsprofiler med hver av de oppløste reflektorenes posisjon i tid og amplitude genereres.

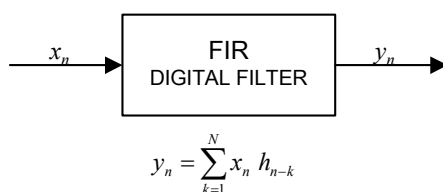
Når et radarmål skal bygges opp kunstig, så kan dette gjøres på flere måter. Både analog og digital teknologi kan benyttes, men når et visst antall programmerbare radarrefleksjoner skal genereres, så vil den digitale måten være å foretrekke pga større fleksibilitet og mindre problemer med miniatyrisering.

Den digitale fremgangsmåten kan enten utføres ved hjelp av direkte realisering eller ved hjelp av beregnet realisering, ref [1]. Direkte realisering er en metode hvor man konstruerer forsinkelse, frekvens- og amplitudeforandring på en måte som direkte etterligner de fysiske prosessene ved refleksjon. Denne måten blir imidlertid fort krevende og med stort volum når mange reflektorer skal realiseres. Figur 5.1 viser en skisse av en jammer med direkte realisert modellering for høyoppløselig radar.



Figur 5.1 Jammer med direkte realisert modellering for høyoppløselig radar

En annen fremgangsmåte er å tenke på refleksjonsprosessen som et digitalt filter. Hvis radarsignalet er inngangssekvensen x_n og refleksjonspunktene er konstantene i impulsresponsen h_n blir utgangssekvensen y_n det ønskede jammesignal, ref Figur 5.2.



Figur 5.2 Generelt digitalt Finite Impulse Response (FIR) filter

Impulsresponsens konstanter h_1, h_2, \dots, h_n er komplekse konstanter gitt den amplitude og fasemodulasjon som hver enkelt reflektor skal påføre radarsignalet. Filterets samplerate fastsetter minste avstand mellom refleksjonene og kun ved de avstandene som representerer en refleksjon vil det være en konstant ulik null i impulsresponsen. På denne måten vil hovedvekten av konstantene ha verdien null, men siden impulsresponsen består av svært mange sampler vil det totale antall konstanter være stort. Strukturen i Figur 5.1 er en parallell realisering av et FIR-filter, som kun tar hensyn til de filterkonstantene som er ulik null.

Beregningen av et standard FIR-filters utgangssekvens gjøres på vanlig måte for digitale filtre, ved hjelp av konvolusjon mellom filterets impulsrespons og inngangssekvensen:

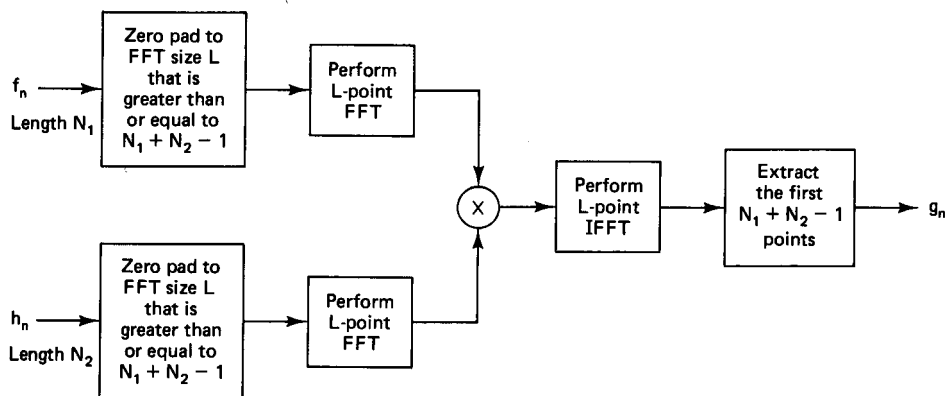
$$y_n = \sum_{k=1}^N x_n h_{n-k} = \sum_{k=1}^N h_n x_{n-k} \quad (5.1)$$

hvor x_n og h_n er to sekvenser med lengde N . Resultatsekvensen y_n får lengden $2N-1$. Når sekvenslengdene blir store blir utregningen av konvolusjonen svært regnekrevende, ref [11]:

$$\begin{aligned} \text{Komplekse multiplikasjoner} &: N^2 \\ \text{Komplekse addisjoner} &: (N-1)^2 \end{aligned} \quad (5.2)$$

Som uttrykkene viser øker antall komplekse regneoperasjoner kvadratisk med sekvenslengdene.

En allment kjent implementasjon for rask konvolusjon er *Fast Convolution Processing* (FCP), ref [11], som benytter FFT-implementasjonen for diskret fourier transform (DFT) som byggekloss. Figur 5.3 viser blokkdiagram for denne metoden.



Figur 5.3 Blokkdiagram for Fast Convolution Processing (FCP), ill. fra [11]

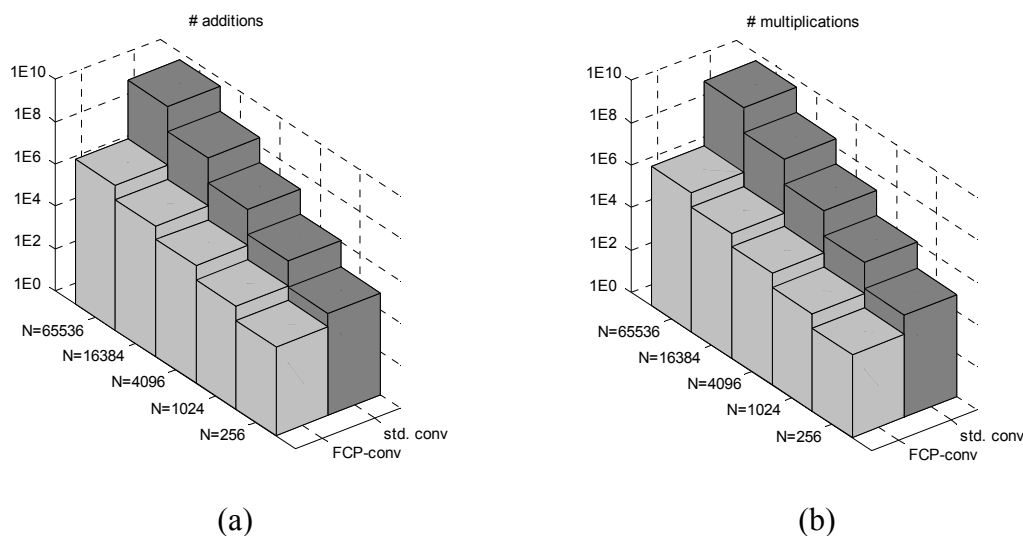
Algoritmen krever at inngangssekvensene nullfylles til lengde L som må være større eller lik N_1+N_2-1 hvor N_1 og N_2 er inngangssekvensenes lengde. Lengden L må være en lovlig størrelse for FFT-funksjonen, dvs $L=2^m$ hvor m er et heltall. Beregningsbelastningen blir her:

$$\begin{aligned} \text{Komplekse multiplikasjoner} &: 3L/2 \log_2 L + L \\ \text{Komplekse addisjoner} &: 3L \log_2 L \end{aligned} \quad (5.3)$$

Figur 5.4 viser antall komplekse beregninger langs z -aksen for noen utvalgte verdier av N langs y -aksen. Figur 5.4 (a) illustrerer antall addisjoner for FCP-konvolusjon sammenlignet mot standard konvolusjon, mens Figur 5.4 (b) viser det tilsvarende for antall multiplikasjoner.

Hver av de komplekse multiplikasjonene krever 4 reelle multiplikasjoner og 2 reelle addisjoner, mens en kompleks addisjon krever 2 reelle addisjoner. Figur 5.4 viser at allerede ved korte sekvenslengder krever FCP-konvolusjon fra 1 til 2 størrelsesordener færre

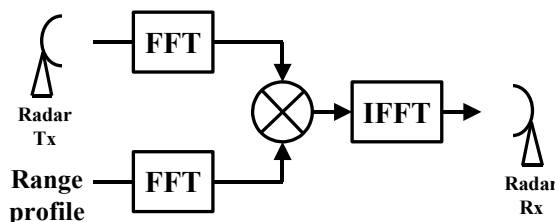
komplekse regneoperasjoner enn standard konvolusjon, og ved lengre sekvenslengder øker forskjellene ytterligere. Dermed er det en stor hastighetsgevinst ved å benytte FCP-konvolusjon.



Figur 5.4 Sammenligning av antall komplekse regneoperasjoner ved FCP-konvolusjon versus standard konvolusjon. (a) – antall addisjoner, (b) – antall multiplikasjoner

Siden direkte realisert modellering som i Figur 5.1 kun tar hensyn til de filterkonstantene som er ulik null, mens beregnet realisert modellering med FCP-konvolusjon som i Figur 5.3 må ta hensyn til alle filterkonstantene, selv om de er lik null, vil det bli et krysningsspunkt for når beregnet realisert modellering vil være mer effektiv enn direkte realisert modellering.

Det er metoden med FCP-konvolusjon som er beskrevet i [1] under betegnelsen *Computed ECM*. Metoden kan også betegnes som modulasjon i frekvensdomenet, da metoden utnytter at inngangssekvensenes spektre kan multipliseres i frekvensdomenet i motsetning til i tidsdomenet hvor inngangssekvensene må konvolveres. Figur 5.5 viser en prinsippskisse for en jammer for høyoppløselig radar som benytter modulasjon i frekvensdomenet:



Figur 5.5 Modell av jammer med beregnet realisering for høyoppløselig radar. (IFFT står for invers FFT)

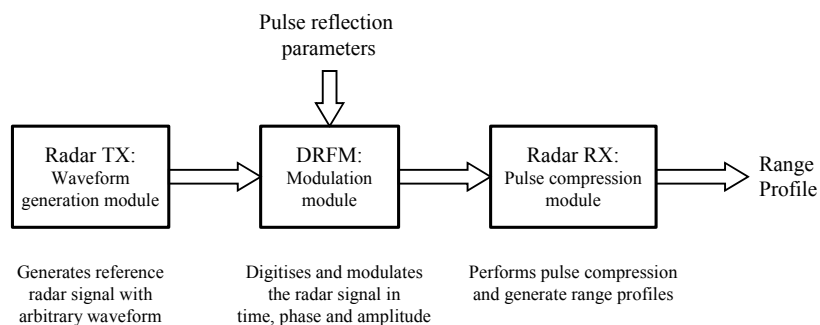
Fordelen med en metode som benytter modulasjon i frekvensdomenet er at jammerens kompleksitet kan holdes nær konstant ved økende antall reflektorer. Derimot vil kompleksiteten øke ved økende antall sampler i mottatt radarsignal og ved økende størrelse

på den dekkede radarscene. Størrelsen på radarscenen som kan jammes til enhver tid vil være gitt av antall datapunkter i FFT-algoritmen minus antall sampler i mottatt radarsignal.

5.3 Datamodell av DRFM-modulator og enkelte radarfunksjoner

Målsettingen med datamodellen er å undersøke hvordan modulatorene i DRFM kan realiseres. Modulatoren skal kunne håndtere påtrykk fra forskjellige radarkilder og skal kunne generere jammesignaler som tolkes som troverdige radarmål etter pulskompresjon i radarmottakere. Datamodellen har på ingen måte noen målsetting om å være en fullstendig modell. Den inneholder kun elementer fra radarens signalbehandlingen som er av interesse for problemstillingen i oppgaven.

Datamodellen er bygget opp ved hjelp av flere moduler og funksjoner. Hovedsaklig består datamodellen av modul for generering av bølgeformer, modul for modulator og modul for pulskompresjon. De forskjellige modulene blir ytterligere supplert av noen tilleggsfunksjoner for å dekke spesielle behov.



Figur 5.6 Skjematisk beskrivelse av datamodell

Som Figur 5.6 viser, er de forskjellige modulene i datamodellen nært knyttet til de fysiske funksjonene eller enhetene i radar og jammer. De enkelte modulene vil bli gjennomgått i mer detalj i de kommende kapitlene. Først gjennomgås generering av radarbølgeformer og pulskompresjon i radarmottakeren, så beskrives jammerens modulator. I denne omgang vil de enkelte realiseringskonseptene for modulatorene bli listet opp, mens en grundigere gjennomgang og diskusjon vil bli omhandlet i kapittel 6. Bakgrunns materialet for implementasjon av datamodellens pulskompresjon er hentet fra [9].

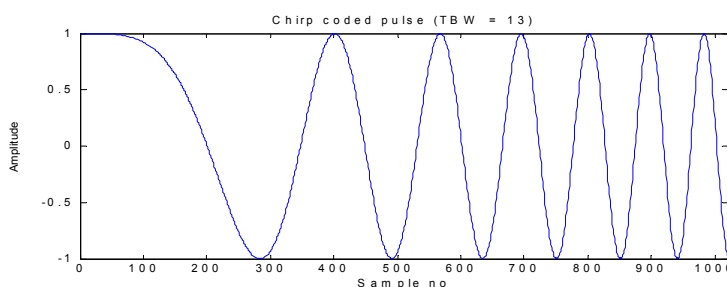
5.4 Modul for generering av radarbølgeformer

Denne modulen tar for seg signalbehandlingen som utføres i forbindelse med generering av radarsignalet. Modulen kan ses på som en programmerbar pulsgenerator med mulighet for forskjellig modulasjon eller koding av bølgeformen.

To hovedgrupper av bølgeformer er implementert i modulen; frekvensmodulerte chirp-pulser og fasekodete pulser.

5.4.1 Chirp

I en chirp-puls blir bærebølgens frekvens variert gjennom pulsen. Mest vanlig er lineær frekvensmodulasjon, mens ulineær frekvensmodulasjon forekommer sjelden selv om den har enkelte fortrinn i radarmottakerens signalbehandling. Figur 5.7 viser en chirp-puls med tid-båndbredde-produkt på 13 som er generert ved hjelp av datamodellen. Matematisk uttrykk for chirp bølgeformen kan finnes i avsnitt 3.2.1.



Figur 5.7 Lineær frekvensmodulasjon

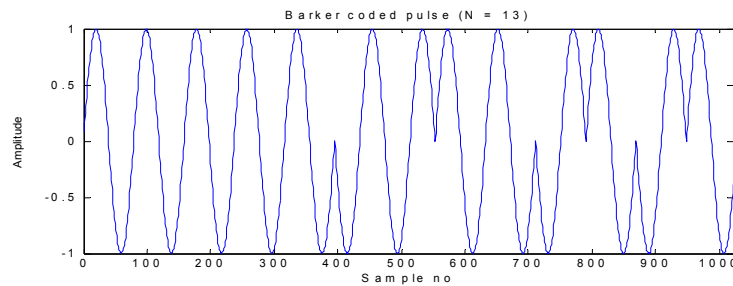
Ved lineær frekvensmodulasjon endres enten frekvensen fra laveste verdi f_1 til høyeste verdi f_2 som i Figur 5.7, eller motsatt vei. Pulsens tid-båndbredde-produkt er gitt av pulsens varighet multiplisert med pulsens båndbredde $B = f_2 - f_1$.

Chirp-pulser har et relativt dårlig sidelobenivå etter pulskompresjon, da høyeste sidelobe ligger på -13 dB i forhold til hovedloben. Dette nivået kan imidlertid forbedres vesentlig med bruk av en vindusfunksjon før pulskompresjonen. Nivåer under -40 dB kan da oppnås.

Bølgeform-modulen har implementert chirp-puls med tid-båndbredde-produkt valgbart i området 3 – 65535. Det er også implementert mulighet for å velge Hamming vindusfunksjon. Andre tid-båndbredde-produkter eller vindusfunksjoner kan legges inn.

5.4.2 Fasekodete pulser

Fasekodete pulser kan ses på som om pulsen er delt inn i N subpulser med lik varighet. Hver av de N subpulsene har en distinkt fase gitt av koden som benyttes. Det er kun binære koder, dvs koder med kun 2 faser som er implementert i bølgeform-modulen. Koder med flere enn 2 faser – polyfase-koder, er også i bruk. Figur 5.8 viser en Barker-kodet puls med pulskompresjon på 13 som er generert ved hjelp av datamodellen. Matematisk uttrykk for fasekodet bølgeform kan finnes i avsnitt 3.2.2.



Figur 5.8 Binær fasekoding

En fasekodet puls har et pulskompresjonsforhold gitt av antall N subpulser og kompresjonsforholdet er i verdi lik tid-båndbredde-produktet til pulsen. De forskjellige fasekodene kan klassifiseres i grupper gitt av deres egenskaper som optimale, maksimumlengde, kvadratiske og komplementære koder. Disse gruppene kan igjen være delt opp i undergrupper. I denne oppgaven er det kun viet oppmerksomhet mot optimale og maksimumlengde koder.

De optimale fasekodene er koder hvis sidelobenivå er optimalt ut i fra kodens lengde og kan derfor finnes for alle kodelengder. De enkelte kodene finnes ved hjelp av dataprogram som søker i gjennom alle mulige koder for å finne den mest optimale. Barker-kodene er en undergruppe av disse, da de kun inneholder koder som har sidelobenivå $1/N$. Således er det kun noen få optimale koder som går inn under kravet til Barker-kodene.

Maksimumlengde koder er kodesekvenser som har maksimum lengde gitt et N -bit tilbakekoblet skiftregister. Kodene blir ofte kalt pseudo-random siden de har en 'nesten tilfeldig' natur og benyttes blant annet også til slumptalls generatorer. Selv om maksimumlengde kodene ikke har optimale sidelobenivåer, så har de som alle fasekodede sekvenser et pulskompresjonsforhold gitt av kodelengden N .

De enkelte fasekodene kan opptre i fire forskjellige *allomorphic forms*, ref [9]. De forskjellige formene er; opprinnelig kode, koden i revers ordning, komplementert kode og komplementert kode i revers ordning. Alle kodeformene har de samme egenskapene når det gjelder sidelobenivå og kompresjonsforhold. Alle bølgeformene i modulen kan velges med de forskjellige allomorphic formene.

Tabell 5.1 viser som et eksempel de forskjellige formene for Barker $N=13$ fasekode.

<i>Allomorphic form</i>	<i>Kodesekvens eksempel</i>
Opprinnelig kode	1111100110101
Revers ordning	1010110011111
Komplementert	0000011001010
Komplementert revers ordning	0101001100000

Tabell 5.1 De forskjellige allomorphic formene

Følgende bølgeformer er implementert i bølgeform-modulen:

<i>Bølgeformens navn</i>	<i>Kodelengder / TBW-produkt</i>
Barker fasekoder	N: 2 - 13
Optimale fasekoder	N: 15, 31, 40
Maksimumlengde fasekoder	N: 3 - 65535
Chirp frekvensmodulasjon	TBW: 3 - 65535

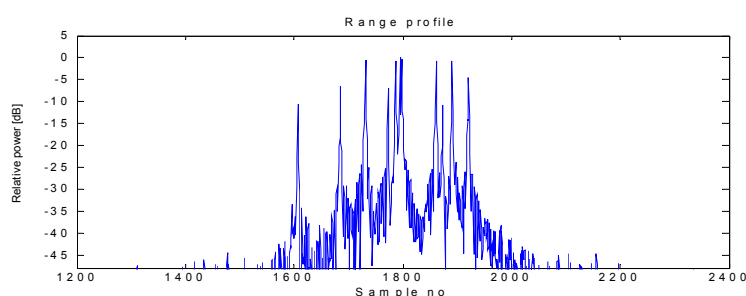
Tabell 5.2 Implementerte bølgeformer i bølgeform-modulen

Eventuelle andre koder kan legges inn.

5.5 Modul for pulskompresjon i radarmottaker

Denne modulen tar for seg pulskompresjonen som utføres i forbindelse med mottak av radarsignalet i radaren. Pulskompresjonen sørger for å komprimere de lange mottatte bølgeformene med pulskoding til et kort komprimert signal. Pulskompresjon utføres ved hjelp av konvolusjon mellom mottatt bølgeform og impulsresponsen til radarmottakerens tilpassede filter. Impulsresponsen til det tilpassede filteret er igjen gitt av den komplekskonjugerte av forventet bølgeform, dvs det utsendte radarsignalet.

Denne modulen benyttes ved verifikasjon av de syntetisk genererte radarrefleksjonene. Pulskompresjonen er i modulen implementert slik at den genererer en avstandsprofil med resultatet fra hvert forsøk. Figur 5.9 viser et eksempel på en avstandsprofil som er generert ved hjelp av modul for pulskompresjon.



Figur 5.9 Eksempel på avstandsprofil generert av modul for pulskompresjon

5.6 Modul for DRFM-modulator

Modul for DRFM-modulator tar for seg signalbehandlingen som gjøres i jammerens modulator ved generering av den virtuelle radarscenen. All signalbehandling gjøres på grunnlag av en kopi av utsendt radarsignal og den ønskede radarscenens impulsresponskoeffisienter. Denne modulen er hoveddelen av datamodellen når det gjelder omfang og innsats.

Syntetisk modellering ved hjelp av både direkte realisering modellering og beregnet realisering modellering er behandlet i modulen. Ved direkte realisering er de enkelte komponentene modellert, mens ved beregnet realisering er forskjellige algoritmer og strukturer modellert. Ved direkte realisering er også alle signalveiene i modulatorene modellert med valgfri oppløsning i antall bit. A/D og D/A-konverterne i DRFM-systemet er også modellert med samme fleksibilitet.

Den direkte realiseringen er brukt som en referanse ved utviklingen av de forskjellige beregnede realiseringene. Hver av realiseringene består av digital signalbehandling i form av varianter og avarter av konvolusjon mellom utsendt radarsignal og impulsresponsens konstanter gitt av den ønskede radarscenen som skal genereres. Avveiningene ligger her i hvilken algoritme eller struktur som skal velges. Av standard algoritmer finnes vanlig konvolusjon og FCP-konvolusjon. I tillegg er flere forskjellige strukturer modellert, hvor det ene ytterpunktet består av en standard konvolusjon som dekker hele radarscenen, mens det andre består av mange FCP-konvolusjoner, et sett for objektene og et sett for bakgrunnen, hvor hver kun dekker en del av radarscenen.

Tabell 5.3 viser en oversikt over alle variantene som er modellert og hvor en detaljert beskrivelse og analyse kan finnes.

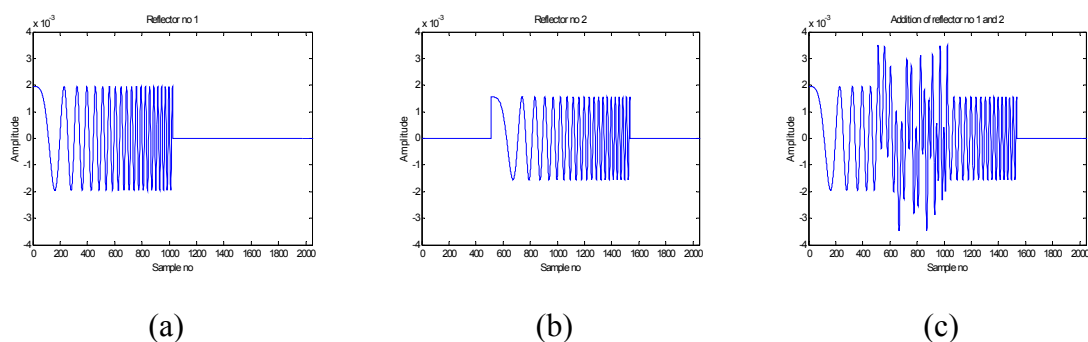
Modul for DRFM-modulator har implementert de forskjellige realiseringskonseptene slik at datamodellen kan være et instrument under utvelgelsen for den digitale modulatorene. I denne prosessen er viktige parametere; maksimal modulatorforsinkelse, akseptabel avstandsområdedekning og total kompleksitet.

Følgende strukturer for DRFM-modulatoren er modellert:

<i>Implementerte algoritmer / strukturer</i>	<i>Kapittel henvisning</i>
Direkte realisering	
- Referansestruktur	6.4
Beregnet realisering	
- Standard konvolusjon	6.5
- FCP-konvolusjon	6.6
- FCP-konvolusjon med separat modulasjon av objekt og bakgrunn	6.7
- Parallele FCP-konvolusjoner	6.8
- Distribuerte FCP-konvolusjoner	6.9
- Distribuerte FCP-konvolusjoner med separat modulasjon av objekt og bakgrunn	6.10
- Distribuerte FCP-konvolusjoner med parametrisk generator for bakgrunnsscenen	6.11

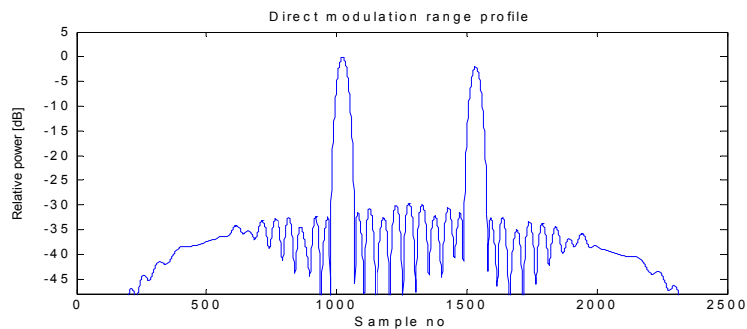
Tabell 5.3 Implementerte strukturer i modul for DRFM-modulator

Figur 5.10 viser et eksempel på direkte realisert modellering hvor innkommende radarsignal, en chirp-puls (TBW = 40) blir modellert av to reflektorer i modulatorene. Figur (a) viser reflektor 1 med forsinkelse 0 sample og amplitude 0 dB, figur (b) viser reflektor 2 med forsinkelse 511 sample og amplitude -2 dB i forhold til reflektor 1, og figur (c) viser resultatsignalet etter summering av de to reflektorene.



Figur 5.10 Resultatsignalet (c) etter modellering av to enkle reflektorer (a),(b)

Figur 5.11 viser avstandsprofilen etter pulskompresjon av resultatsignalet i Figur 5.10. Legg merke til at reflektorene er forskjøvet en pulsbredde dvs 1023 sampler til høyre etter kompresjonen. Dette skyldes konvolusjonens natur, der utgangssekvensen har en utbredelse lik summen av de to inngangssekvensenes lengde.

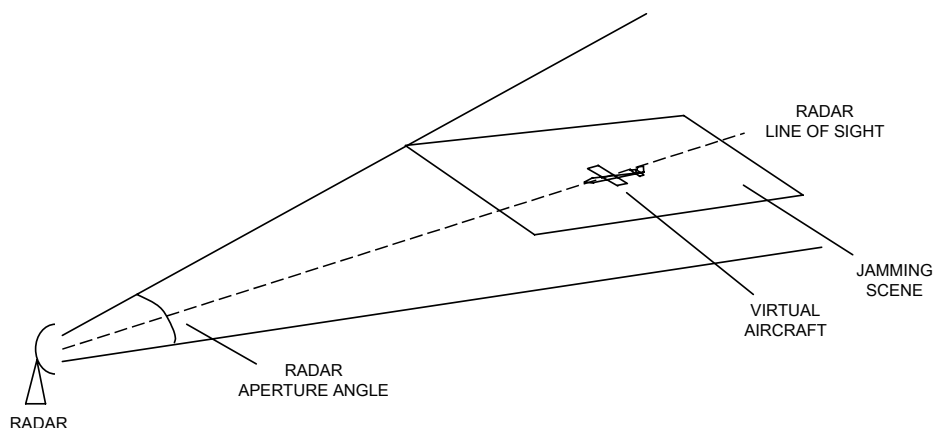


Figur 5.11 Eksempel på avstandsprofil etter pulskompresjon av resultatsignalet i Figur 5.10 generert ved hjelp av direkte modulasjon

5.7 Definisjon av objekt med romlig utstrekning for test av datamodellen

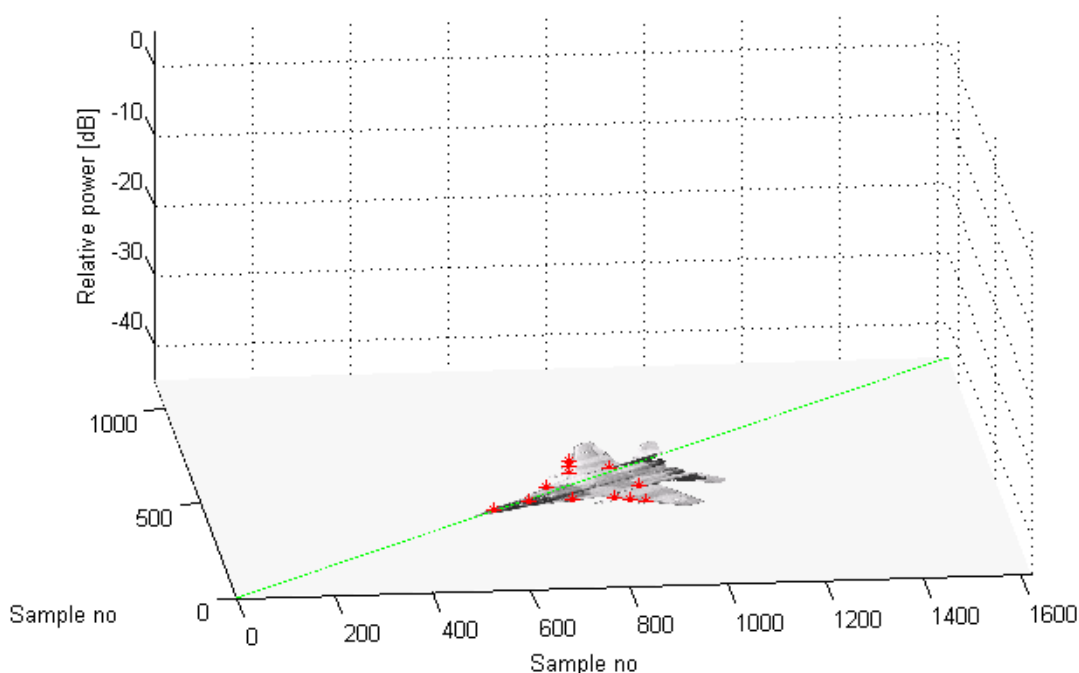
Objektet er valgt med henblikk på å vise radarens mulighet til å løse opp et komplekst radarmål ved forskjellige bølgeformer. Objektet er et jagerfly hvor det er definert refleksjonspunkter som typisk vil kunne reflektere en innkommende bølgeform. Et virkelig jagerfly vil antakeligvis ha et stort antall refleksjonspunkter, men som et eksempel er det her valgt kun 12 reflektorer. Antall reflektorer og deres posisjon, amplitude og fase er definert i datamodellen.

Datamodellen er bygget opp rundt et tenkt scenario hvor radaren belyser en scene der jammeren genererer et virtuelt objekt i form av et jagerfly. Hele scenen forutsettes å bli belyst av radaren under modelleringen. Figur 5.12 viser en skisse som klargjør scenariet.



Figur 5.12 Radarens virtuelle scene med et falskt radarmål

Datamodellen genererer scenen, ref Figur 5.13, med ønsket størrelse i x og y-retning, samt avstandsprofilens relative amplitude i z-retning. En grønn linje indikerer radarens siktelinje, mens refleksjonspunktene er merket med røde stjerner på det virtuelle objektet som er tegnet på xy-planet i figuren. Radarens siktelinje og det virtuelle objektets orientering med de tilhørende refleksjonspunktene kan fritt roteres i xy-planet, dvs objektens aspektvinkel mot radaren kan endres. I de påfølgende eksemplene er flyet belyst forfra med en offset på 7° for å skille de enkelte refleksjonspunktene fra hverandre.



Figur 5.13 Datamodellens virtuelle radarscene

De enkelte refleksjonspunktene blir projisert ned på radarens siktelinje og vektorielt summert sammen i datamodellen. Avstandsprofilen genereres deretter på grunnlag av bølgeformens belysning av refleksjonspunktene. Refleksjonspunktenes koordinater og tilhørende amplituder er vist i Tabell 5.4.

<i>X-koordinat</i> <i>[sample no]</i>	<i>Y-koordinat</i> <i>[sample no]</i>	<i>Amplitude</i> <i>[dB]</i>
584	412	-10
662	454	-3
750	454	-6
708	525	-6
775	657	-3
772	632	-3
764	600	0
838	458	0
868	447	-3
898	440	-3
897	518	-10
850	618	-10

Tabell 5.4 Refleksjonspunktene posisjon og amplitude i testobjektet

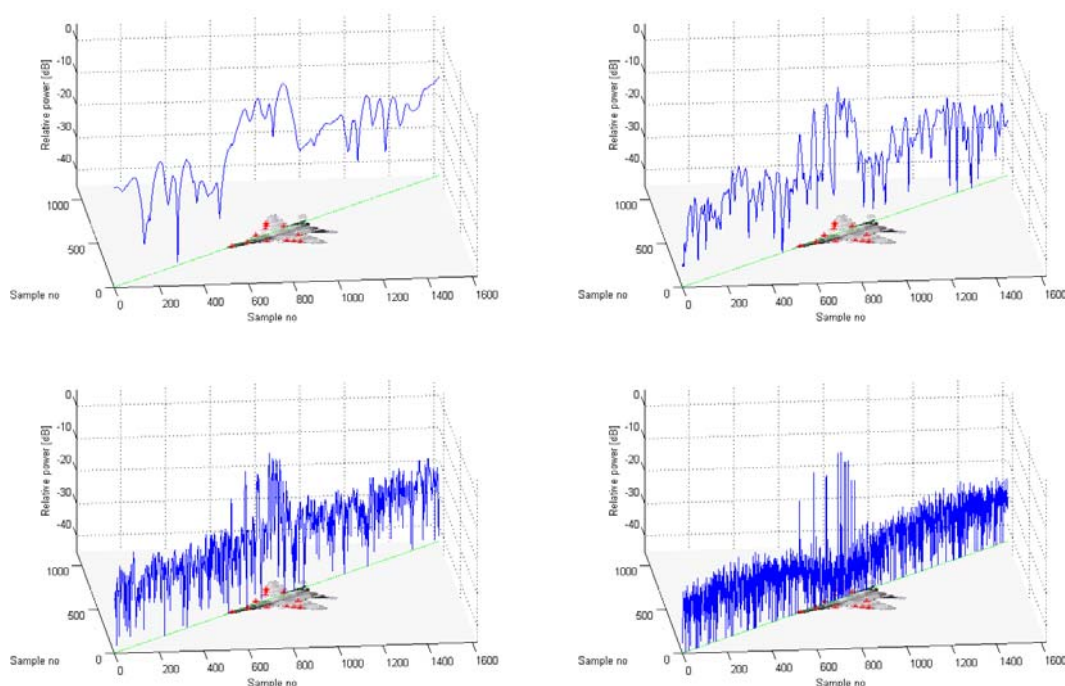
Etter pulskompresjonsberegningene plottes avstandsprofilen inn i jammescenen med relativ amplitude i z-aksen og med radarens siktelinje som x og y-koordinater. Dermed blir bildet av objektet som ligger i xy-planet overlappet med dens egen avstandsprofil langs radarens siktelinje.

5.8 Eksempel på modellering av radarrefleksjoner for HRR-radar

Modelleringen er utført med testobjektet mot forskjellige bølgeformer og kompresjonsfaktorer. Først vises forskjellige HRR-avstandsprofiler med maksimumlengde fasekodet bølgeform.

Figur 5.14 viser forskjellige HRR-avstandsprofiler som er prosessert med kompresjonsgrad på 15, 63, 255 og 1023. Lavest kompresjonsgrad øverst til venstre og høyest kompresjonsgrad nederst til høyre i figuren. Alle bølgeformene har lengde 1023 sampler og benytter maksimumlengde sekvens fasekodning.

Som Figur 5.14 viser, så har avstandsprofilene et relativt høyt sidelobenivå etter pulskompresjon. Dette skyldes de valgte bølgeformenes begrensende egenskaper til sidelobeundertrykking, slik at pulskompresjonsprosessen danner sidelober i tidsdomenet, både før og etter refleksjonspunktene i avstandsprofilene. Ved store krav til lave sidelober vil typisk andre bølgeformer bli foretrukket under konstruksjonen av radarsystemene.

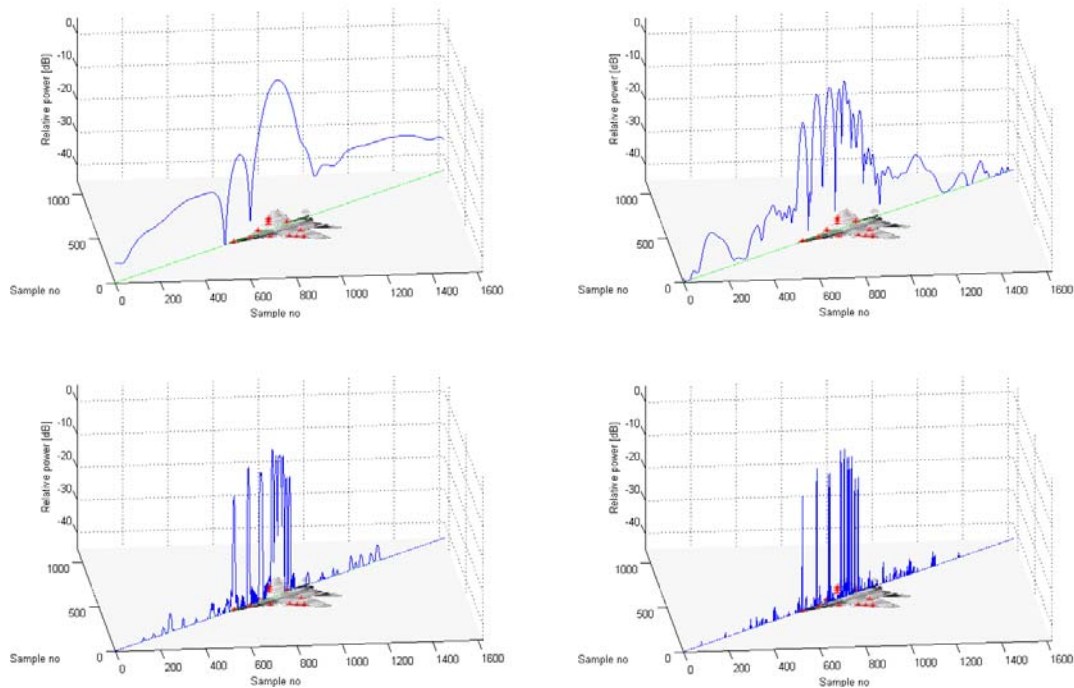


Figur 5.14 HRR-avstandsprofiler med maksimumlengde fasekodet bølgeform og forskjellig grader av kompresjon

Hvis bølgeformene ikke hadde benyttet kompresjon ville dens oppløsning i avstand være gitt av bølgeformens lengde som i disse tilfellene ville være lik 1023 samplere. Med den høyeste kompresjonen derimot, har bølgeformen en oppløsning på 1 sample. Her blir så godt som hver eneste reflektor løst opp og er avskilt fra de andre reflektorene. Deres posisjon og amplitude gjenspeiler godt testobjektet som er grunnlaget for modelleringen. Når kompresjonsgraden senkes vil objektets refleksjonspunkter mer og mer smøres sammen og tilslutt kan ingen av de enkelte reflektorene skilles fra hverandre. Både reflektorenes posisjon og amplitude blir usikker.

Når forsøket gjentas, Figur 5.15, med chirp bølgeformer med samme kompresjon som i forrige forsøk, viser det seg at avstandsprofilene nå har langt lavere sidelobenivå. Chirp bølgeformen har i utgangspunktet ikke lavere sidelober enn -13dB, men ved hjelp av Hamming vindusfunksjon på mottatt radarsignal før pulskompresjonen er sidelobenivået senket til under -40dB.

Siden både Figur 5.14 med fasekodete signaler og Figur 5.15 med chirp-kodete signaler har de samme kompresjonsgradene, kan det ses at reflektorenes posisjon og oppløsning er omtrentlig like i de to figurene. Dette ses best ved de høyere kompresjonsgradene, da de høye sidelobene vanskeliggjør sammenlikning ved lav kompresjon.



Figur 5.15 HRR-avstandsprofiler med chirp bølgeform og forskjellig grader av kompresjon

Dermed kan det fastslås at bølgeformens koding ikke er avgjørende for bølgeformens evne til å løse opp de enkelte reflektorene i et komplekst radarmål. Det er ene og alene bølgeformens kompresjonsfaktor foruten bølgeformens varighet som avgjør dette.

I dette eksemplet er størrelsen på den tenkte radarscene 1620 x 1150 sampler. Radarsignalet som modulatorene mottar har en varighet på 1023 sampler, mens impulsresponsens koeffisienter har en varighet av 850 sampler. Maksimalt kan koeffisientene ha en lengde av 1026 sampler i denne konfigurasjonen, slik at den genererte avstandsprofilen totalt får en lengde på 2048 sampler.

Hvis det ønskes å plassere reflektorer utover området på 1026 sampler, kan dette gjøres på to måter; øke konvolusjonens FFT-transformasjon til neste standard FFT-størrelse eller innføre en konvolusjon til, tidsforskjøvet i parallell med den forrige konvolusjonen. Om modulatorens innskuddsforsinkelse kan økes uten at dette fører til problemer i forhold til maksimal forsinkelse for troverdig jamming hos aktuell radartype, så kan en større FFT-transformasjon velges. Ellers må en struktur med flere konvolusjoner innføres, ref kapittel 6.8 og 6.9.

Regneeksemplet overfor kunne benyttet tiden mellom hver pulsutsendelse, dvs radarens pulsrepetisjonsintervall (PRI) som mål for hvor stor innskuddsforsinkelsen maksimalt kunne være. Når forsinkelsen er lik en PRI vil en radarpuls som lagres i DRFM, først bli benyttet som jamming under neste pulsutsendelse fra radaren. Ønskemålet er at modulatorene er så raske at jammingen kan brukes i inneværende PRI, men avhengig av radartypen og hvor den virtuelle scenen skal plasseres kan f.eks maksimal forsinkelse fra 1 til 3 – 4 ganger PRI godtas.

6. Analyse av mulige realiseringskonsepter for modulatorene

6.1 Innledning

Dette kapitlet tar for seg de forskjellige realiseringskonseptene for bruk mot HRR-radar som er utviklet for modulatorene under hovedfagsarbeidet. Modellering av HRR-avstandsprofiler danner grunnlaget for all modellering av komplekse refleksjoner i alle avbildende radarer. Dermed vil også konklusjonene fra dette kapittel være gyldige mot de mer avanserte SAR og ISAR-radarene.

De forskjellige forslagene til beregnet realisering blir sammenlignet mot dagens direkte realiserede metode, som kan ses på som en referansestruktur i denne sammenheng. På denne måten blir de forskjellige realiseringene funksjonelt verifisert.

Analysen bygger på resultater oppnådd ved hjelp av datamodellen beskrevet i kapittel 5. Hvert enkelt forslag til realisering blir detaljbeskrevet ved hjelp av strukturdiagram og eksempler på genererte avstandsprofiler.

6.2 Krav for valg av realiseringsmetode for digital modulator

Formålet med arbeidet er å sammenligne kjente realiseringskonsepter for en digital modulator, samt å finne fram til nye konsepter som kan være mer egnet i denne sammenheng. Målsettingen er å finne en metode som innfører minst mulig innskuddsforsinkelse gjennom modulatorene, samtidig som den dekker et akseptabelt avstandsområde. Det er ønskelig om modulatorene både kan modellere de komplekse refleksjonene fra forskjellige objekter og til en viss grad kunne håndtere modellering av bakgrunnen i radarscenen.

Hva som er akseptable verdier for innskuddsforsinkelse og størrelsen på avstandsområdet er helt avhengig av det gitte radarsystemet. Siden den digitale modulatorene er tenkt brukt mot høyst forskjellige radarer, er det viktig at dens oppbygning med rekonfigurerbare FPGA-kretser utnyttes til det fulle. Det vil bli gitt eksempel på hvordan modulatorens krav er avhengig av radartype og hvordan dette håndteres i konfigurasjonen i aktuell FPGA i modulatorene.

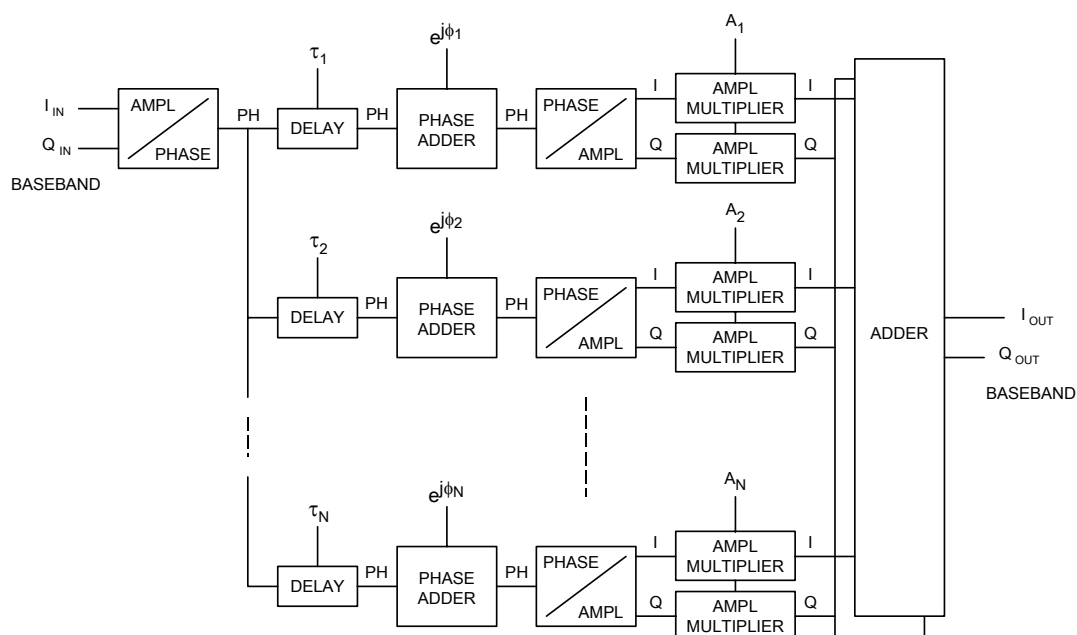
6.3 Definisjon av referansesignal

Analysen er utført ved hjelp av to forskjellige referansesignaler. Referanse 1 er den samme modellen som ble bruk i kapittel 5.7. Disse 12 reflektorene utgjør typiske refleksjonspunkter i et komplekst radarmål i form av et virtuelt jagerfly. Når denne referansemodellen benyttes i dette kapitlet, vil den bli belyst av en chirp bølgeform med $TBW=127$.

Referanse 2 består av 31 reflektorer, hver med en amplitudforskjell på -1 dB og med en avstand på 80 sample, fra forrige reflektor. Total dynamikk blir dermed 30 dB. Referanse 2 har et helt kunstig refleksjonsmønster og er kun ment for å visuelt forklare de enkelte strukturenes virkemåte og ytelse. Referanse 2 vil bli belyst av en chirp bølgeform med $TBW=1023$.

6.4 Referansestruktur – direkte realisering

Som nevnt i kapittel 5.2 er strukturen med direkte realisering en direkte overgang fra den fysiske verden med modulasjon i tid, amplitude og fase separat for hver av de enkelte reflektorene. Resultatet fra alle reflektorene blir tilslutt summert sammen. Figur 6.1 viser detaljert strukturdiagram av modellen for direkte realisering.



Figur 6.1 Detaljert strukturdiagram for modulator med direkte realisering

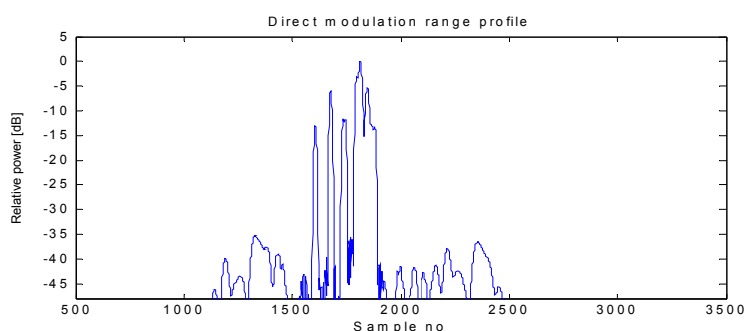
Det detaljerte strukturdiagrammet i Figur 6.1 har opprinnelse fra FFIs eksperimentelle radarjammer EKKO II, hvor denne strukturen er implementert med beregnet kapasitet. Den fulle strukturen skal blant andre strukturer implementeres i den digitale modulatoren. For å realisere en effektiv fasemodulator er det her valgt å foreta en konvertering fra kartesiske til polare koordinater hvor kun fasevinkelen føres videre, dvs det er underforstått at signalet har amplitude lik 1. Dette fasekodete signalet blir påført forsinkelsen τ og ført videre til

faseaddereren som adderer til ønsket fasemodulasjon. Alternativt kunne fasemodulasjonen utføres direkte i det kartesiske planet, men med tyngre beregninger til følge.

Før amplitudemodulasjonen utføres blir det fasekodete signalet tilbakeført til det kartesiske system i form av amplitudesignalene I og Q. Selve amplitudemodulasjonen foretas ved hjelp av to multiplikatorer, en for hver av I og Q-signalene. Tilslutt blir alle I-signalene og alle Q-signalene summert sammen.

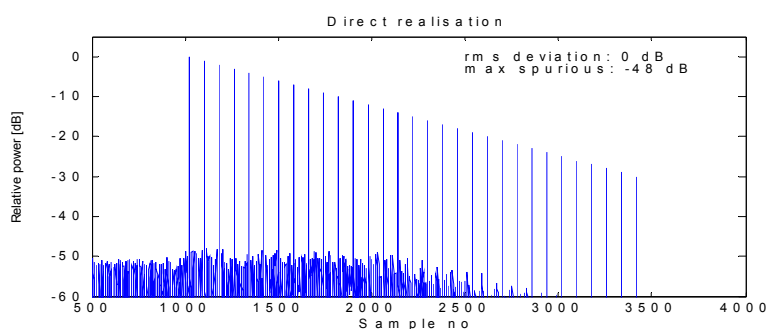
Konverteringene frem og tilbake mellom kartesiske og polare koordinater er valgt grunnet implementering i FPGA hvor små oppslagstabeller er tilgjengelige. Disse tabellene er effektive og vil lett kunne inneholde konverteringer mellom koordinatsystemer. På samme måte kan en oppslagstabell inneholde en amplitudemodulator, riktignok med relativt grov oppløsning.

Figur 6.2 viser et komplekst radarmål, referanse 1, med 12 reflektorer som er generert ved hjelp av direkte modulasjon i datamodellen. Bølgeformen er her en chirp med TBW=127.



Figur 6.2 Avstandsprofil etter pulskompresjon av et komplekst radarmål (referanse 1) generert ved hjelp av direkte realisering

Figur 6.3 viser referanse 2 med 31 reflektorer og chirp bølgeform med TBW=1023.



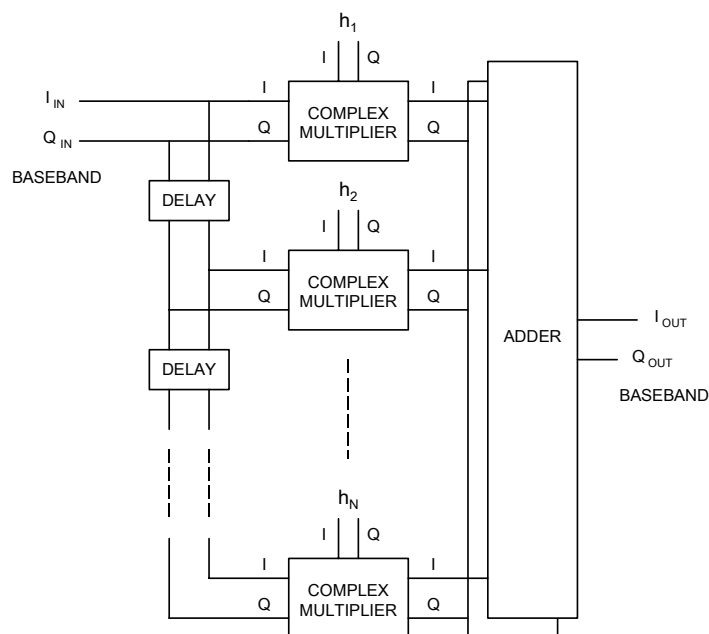
Figur 6.3 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av direkte realisering

Figurene ovenfor viser at direkte realisering genererer den ønskede avstandsprofil etter pulskompresjon. Siden referanse 2 beregnes til gjengivelse med 0 dB rms-avvik og med -48 dB maksimal spuriøsnivå, så må direkte realisering anses til å være en god referanse under utvikling av nye realiseringer. At det maksimale spuriøsnivået er -48 dB skyldes i hovedsak

den belyste bølgeformens egenskaper for sidelobeundertrykking og er således ikke forbundet med realiseringen av modulasjonen.

6.5 Standard konvolusjon

Den mest rett fram metoden å realisere standard konvolusjon på er ved hjelp av et standard FIR-filter. Utgangssignalet fra et FIR-filter kan beregnes ved hjelp av konvolusjon mellom inngangssignalet og filterets impulsrespons. Figur 6.4 viser et FIR-filter oppbygget i tradisjonell konfigurasjon. Hver av de komplekse multiplikatorene beregner produktet mellom tilhørende impulsresponskoeffisient og en forskjøvet kopi av inngangssignalet. Tilslutt summeres resultatet.

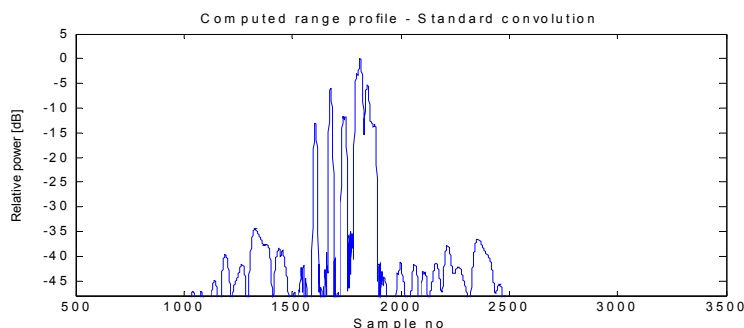


Figur 6.4 Detaljert strukturdiagram for modulator realisert med standard konvolusjon

Resultatmessig blir utgangssignalet ut fra FIR-filteret identisk med signalet fra den direkte realiseringen i Figur 6.1. Implementasjonsmessig er det imidlertid noen forskjeller mellom Figur 6.1 og Figur 6.4. Men, ved å splitte opp kompleks multiplikator i Figur 6.4 i en fasemodulator og en amplitudemodulator, og ved å innføre konvertere mellom kartesiske og polare koordinater, så er implementasjonene kompatible. Dette bekrefter at direkte realisering er det samme som beregnet realisering ved hjelp av FIR-filter. Dette forutsetter imidlertid at forsinkelsesblokkene kan programmeres til ønsket forsinkelse. I vanlig FIR-filter tankegang er det forutsatt at forsinkelsene er av fast varighet, lik samplertaten. Hvis modulatorene skal implementeres på denne måten vil kompleksiteten øke kraftig og vil antakeligvis kun egne seg til komplekse refleksjon innenfor et begrenset avstandsområde.

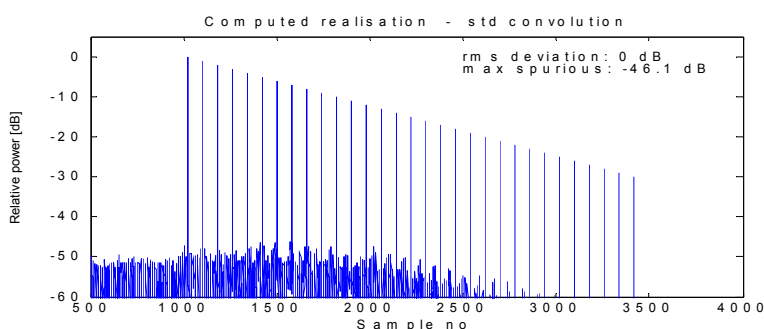
Figur 6.5 viser et komplekst radarmål, igjen referanse 1, men denne gangen generert ved hjelp av standard konvolusjon. Forskjellen mellom direkte realisering – Figur 6.2, og beregnet realisering ved hjelp av standard konvolusjon – Figur 6.5, kan ses som mindre

avvik i de lavere sideløber. Dette skyldes hovedsaklig forskjellig oppløsning under beregningene, da den direkte realiserede strukturen er datamodellert med aktuell dataordbredde, mens de beregnede realiseringene er datamodellert internt med flyttall grunnet programmeringstekniske begrensninger.



Figur 6.5 Avstandsprofil etter pulskompresjon av et komplekst radarmål (referanse 1) generert ved hjelp av standard konvolusjon

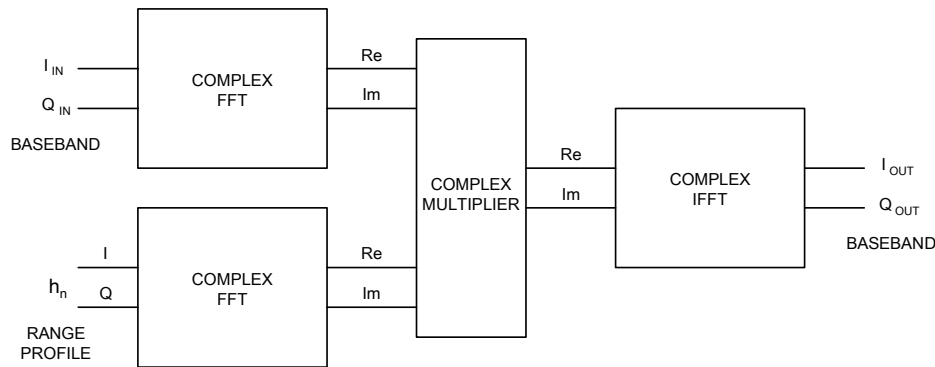
Figur 6.6 viser at standard konvolusjon genererer avstandsprofilen av referanse 2 med god nøyaktighet, 0 dB avvik. Som nevnt vil et mindre avvik i de lavere sideløbene gi seg utslag i et noe høyere maksimum spuriøsnivå på -46,1 dB.



Figur 6.6 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av standard konvolusjon

6.6 FCP-konvolusjon

FCP er en implementasjon av konvolusjon som baserer seg på FFT-implementasjonen av DFT. Dens store fortrinn er dens hastighet, spesielt ved økende lengde i inngangssekvensene. FFT øker kompleksiteten med $\log_2 N$ mot standard konvolusjons N^2 , ref ligning 5.2 og 5.3, hvor N er antall sampler i sekvensen. Figur 6.7 viser dens struktur.



Figur 6.7 Detaljert strukturdiagram for modulator realisert med FCP-konvolusjon

Et annet punkt å legge merke til er at om h_n er kjent før modulasjonstidspunktet så kan transformasjonen av h_n til frekvensdomenet utføres separat og lagres på forhånd. Dermed kan denne delen av problemet tas ut fra den tidskritiske implementasjonen. Det som da står igjen er en kompleks FFT, en kompleks multiplikasjon og en kompleks invers FFT.

FFT-transformasjonene finnes i flere implementasjoner for FPGA med varierende ytelse, vanligvis fra 1 til 6 klokkesykluser pr datapunkt. Implementasjonene er enten bygget opp med tre faser hvor samplene leses inn, prosesseres og leses ut, eller med en *pipeline* hvor samplene flyter igjennom en prosesseringsstruktur fra inngang til utgang.

FFT-transformasjonene har like mange utgangssampler i frekvensdomenet som inngangssampler i tidsdomenet. I alle implementasjonene klokkes utgangssamplene ut på serieform, men deres rekkefølge kan variere noe avhengig av implementasjon. Den serielle utgangen passer fint mot den etterfølgende komplekse multiplikatoren som dermed kan multiplisere de to frekvensspektra serielt med kun en kompleks multiplikator. På samme måte skal resultatspektrumet fra multiplikatoren klokkes serielt inn i den etterfølgende invers FFT.

Forsinkelsen gjennom modulatoren beregnes fra første sample av radarsignalet er mottatt til første sample av jammesignalet sendes ut. Forsinkelsen gjennom modulatoren blir dermed sterkt avhengig av antall datapunkter i FFT-transformasjonene, men uttrykkene avviker noe avhengig av om implementasjonen er *pipelined* eller ikke. Følgende uttrykk er utviklet for *non-pipelined* implementasjon, blant annet ved hjelp av FFT data i [28].

$$t_d = \frac{N_{FFT}(2 + 2K) + M}{f_{clk}} \quad (6.1)$$

hvor N_{FFT} er antall datapunkter i FFT-transformasjonene, K er implementasjonsavhengig konstant for antall klokkesykluser pr datapunkt i FFT, M er implementasjonsavhengig konstant for antall klokkesykluser pr kompleks multiplikasjon og f_{clk} er aktuell klokkefrekvens for modulatoren. Konstanten M vil i FPGA med maskinvarerestøttet multiplikasjon ofte være så lav som 2, da en kompleks multiplikasjon kan bygges opp av 4 reelle multiplikasjoner og 2 reelle addisjoner.

Følgende uttrykk gjelder for *pipelined* implementasjon, utledet av pipeline FFT data fra [33], men kan være noe implementasjonsavhengig:

$$t_d = \frac{t_{IB} + 2t_{FFT} + 2t_{BR} + M}{f_{clk}} \quad (6.2)$$

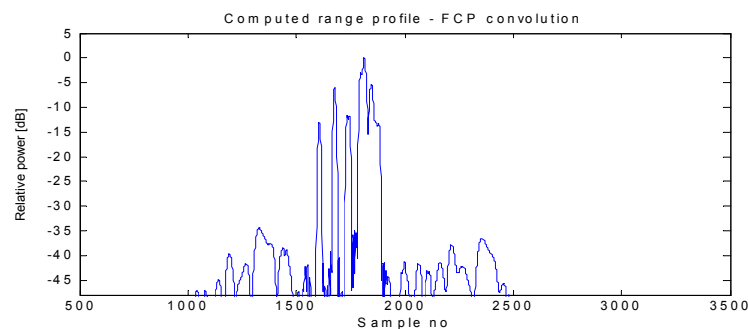
hvor

$$t_{IB} = N_{FFT} / 4 - 1$$

$$t_{FFT} = N_{FFT} / 2 + 8 \log_2(N_{FFT}) - 20$$

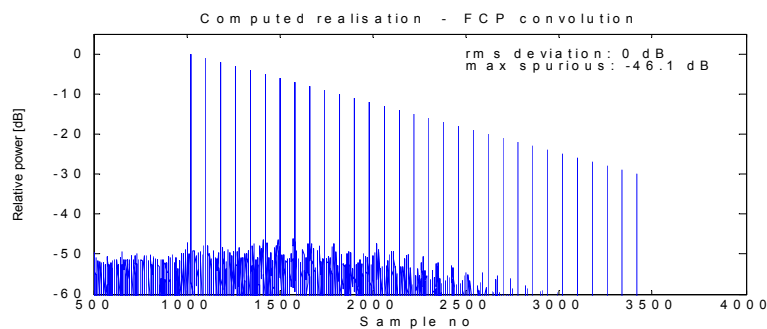
$$t_{BR} = N_{FFT} / 2 - 2^{\lceil \log_2(N_{FFT}/2) \rceil} - 2^{\lceil \log_2((\log_2(N_{FFT}/2)+1)/2) \rceil} + 9$$

hvor N_{FFT} er antall datapunkter i FFT-transformasjonene og M er implementasjonsavhengig konstant for antall klokkesykluser pr kompleks multiplikasjon og f_{clk} er aktuell klokkefrekvens for modulatorene. Konstanten M vil som i ligning (6.1) ofte være lik 2. Mer om FFT implementasjoner kan finnes i kapittel 8.5.



Figur 6.8 Avstandsprofil etter pulskompresjon av et kompleks radarmål generert ved hjelp av FCP-konvolusjon

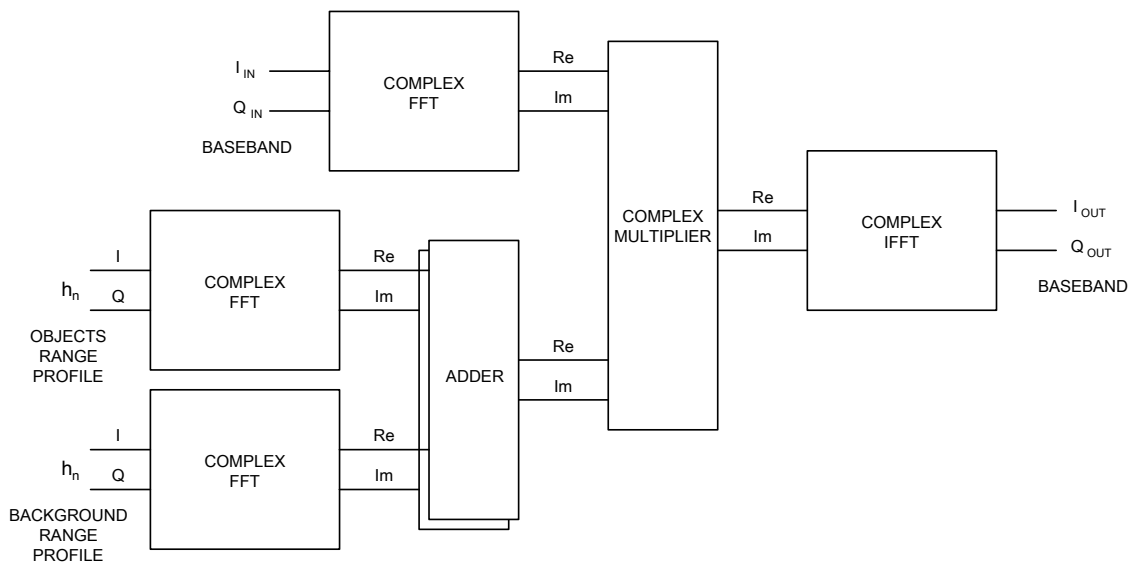
Som Figur 6.8 og Figur 6.9 viser er resultatet identisk om modulatorene er implementert ved hjelp av standard konvolusjon eller FCP-konvolusjon. Både rms-avviket og det maksimale spuriørsnivået er likt.



Figur 6.9 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av FCP-konvolusjon

6.7 FCP-konvolusjon med separat modulasjon av objekt og bakgrunn

Siden modulasjonsdata for de enkelte radarmålene eller objektene gjerne genereres separat fra bakgrunnen i den virtuelle radarscenen, ville det være ønskelig om modulatoren kunne foreta sammenkoblingen av de to datastrømmene under selve modulasjonen. Figur 6.10 viser hvordan dette kan gjøres.

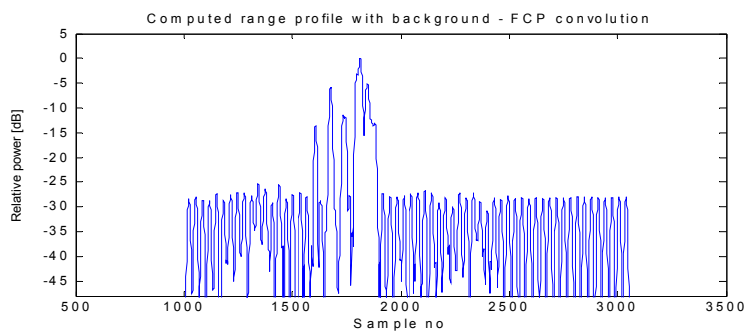


Figur 6.10 Detaljert strukturdiagram for modulator realisert med FCP-konvolusjon med separat modulasjon av objekt og bakgrunn

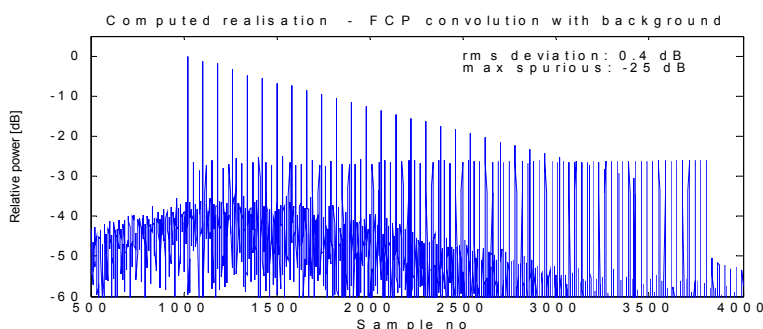
Både objektene og bakgrunnens impulsresponskoeffisienter transformeres til frekvensdomenet ved hjelp av to FFT-transformasjoner. Da FFT-transformasjonene er lineære operasjoner kan frekvensspektrene adderes før videre konvolusjon som ved vanlig FCP-konvolusjon. Som tidligere nevnt, så overføres data fra en standard FFT på serieform slik at addisjonen kan utføres etter hvert som data blir tilgjengelig. Dermed medfører tillegget med separat modulasjon av objektene og bakgrunnen svært lite når det gjelder tidsforbruk, men kompleksiteten øker totalt sett med en kompleks FFT og to reelle addere.

Figur 6.11 viser det tidligere eksemplet, men nå er avstandsprofilen generert med en uniform bakgrunn over 2048 sampler. Det kan ses at bakgrunnens amplitude varierer noe, spesielt i nærheten av reflektorene. Dette skyldes pulskompresjonens sidelober av de sterke reflektorene i scenen.

I praksis vil bakgrunnen bli generert med en del tilfeldige amplituder for å gi inntrykk av 'naturlig' bakgrunn. Dermed vil antakeligvis ikke noen rester fra pulskompresjonen ødelegge det totale inntrykket. For å oppnå en troverdig bakgrunn vil det nok være påkrevd at både den midlere amplituden, variasjonen i amplituden, variasjonens fordeling og periodetid kan endres som funksjon av avstanden.



Figur 6.11 Avstandsprofil med bakgrunn etter pulskompresjon av et komplekst radarmål generert ved hjelp av FCP-konvolusjon



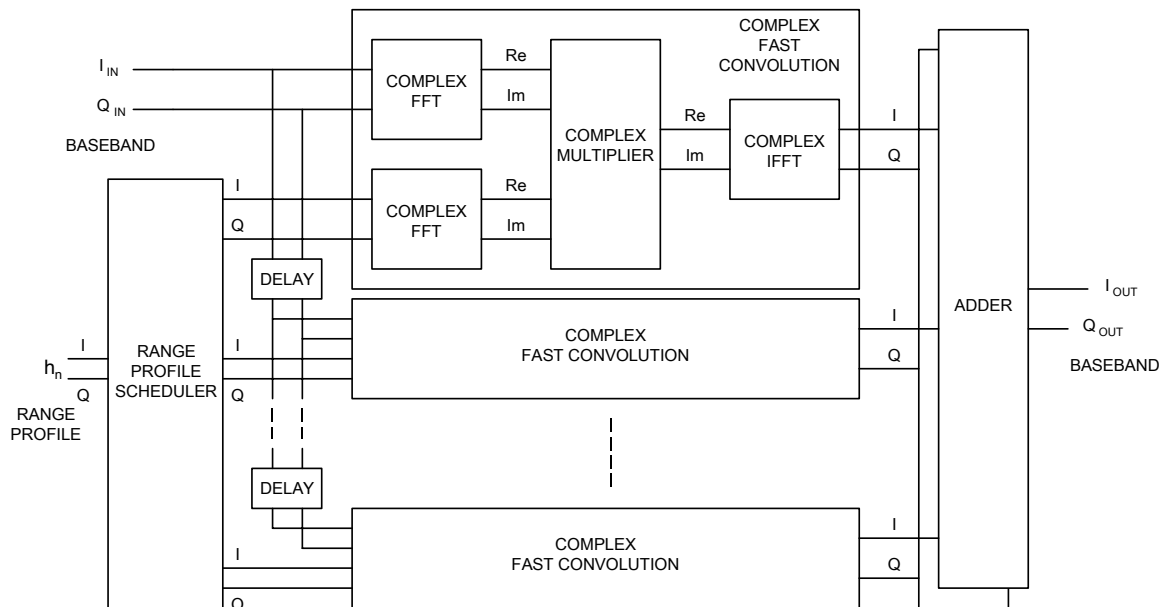
Figur 6.12 Avstandsprofil med bakgrunn etter pulskompresjon av referanse 2 generert ved hjelp av FCP-konvolusjon

Både Figur 6.11 og Figur 6.12 gir et inntrykk av refleksjoner med et naturlig flatt nivå omkring. Figur 6.12 viser også at det generelle støynivået øker som følge av det store antall refleksjoner. Dette merkes best der hvor sidelobeundertrykningen etter pulskompresjonen er svakest.

6.8 Parallele FCP-konvolusjoner

Målsettingen er å holde forsinkelsen gjennom modulatoren på et minimum, samtidig som modulatorens avstandsområde holdes tilstrekkelig stor, noe som er gitt av antall datapunkter i konvolusjonen. En mulig implementasjonsstrategi vil være å kombinere et antall tidsforskjøvede FCP-konvolusjoner slik at hver av konvolusjonene kan være små, og dermed raske, mens deres totale avstandsområde blir tilstrekkelig stort.

Figur 6.13 viser et strukturdiagram for en slik strategi. Figuren indikerer at konvolusjonene er tidsforskjøvede ved at forsinkelser er lagt inn mellom de enkelte konvolusjonsblokkene.

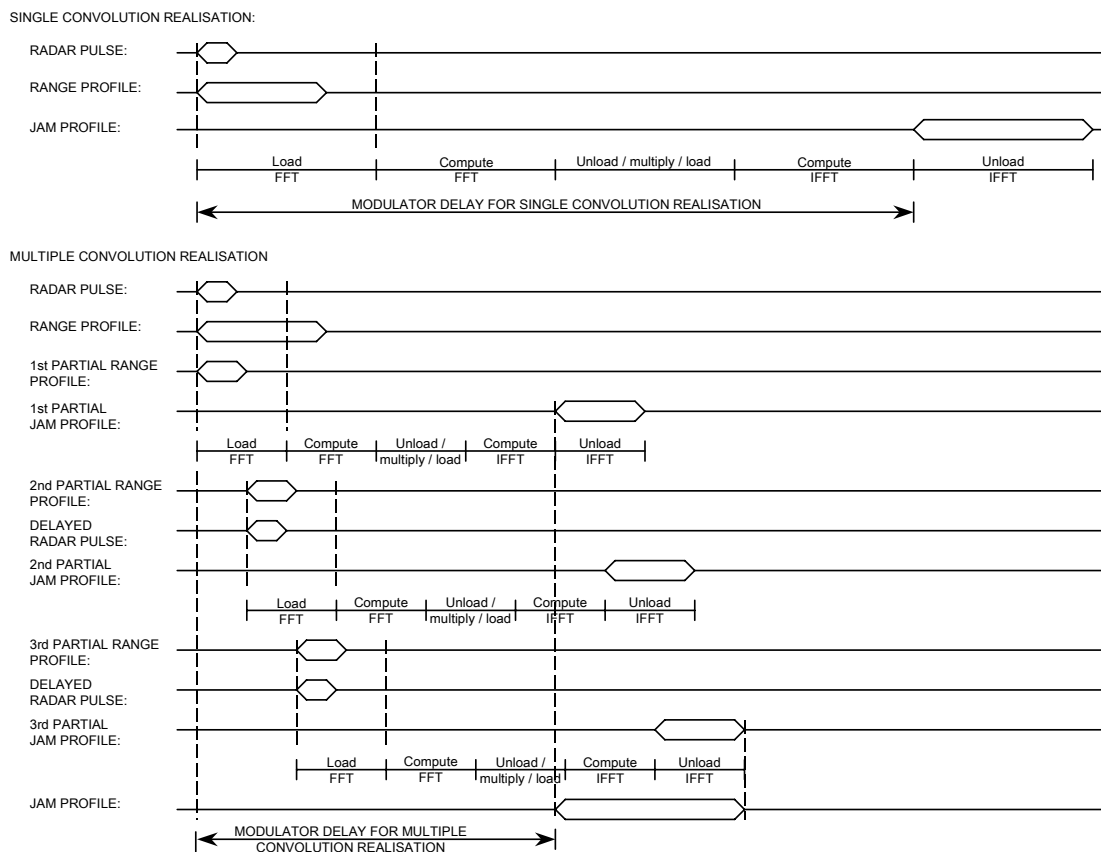


Figur 6.13 Detaljert strukturdiagram for modulator med flere FCP-konvolusjonsblokker

Forsinkelsene mellom de enkelte konvolusjonsblokkene er tilpasset antall datapunkter i konvolusjonen i forhold til det totale avstandsområdet som skal dekkjes. Hvor mange konvolusjonsblokker som trengs til enhver tid kommer an på radarsignalets varighet, hvor stort avstandsområde som skal manipuleres og hva kravet til minimumsforsinkelse gjennom modulatoren er.

I følge kapittel 5.2 må konvolusjonenes FFT-transformasjoner minimum ha lengde lik $N_1 + N_2 - 1$ hvor N_1 og N_2 er sekvenslengdene til henholdsvis radarsignal og impulsresponskoeffisientene. I tillegg må transformasjonen være av en standard FFT-størrelse. Utover minimumskravet, så velges store eller små transformasjoner avhengig av modulatorens aktuelle hastighetskrav. Færre datapunkter i FFT-transformasjonen gir mindre forsinkelse, men også mindre dekning i avstand. For å kompensere dette må antall konvolusjonsblokker økes. Siden konstruksjonen skal implementeres i FPGA, så kan kretsen reprogrammeres og optimaliseres for hvert enkelt bruksområde under bruk.

Blokken *Range profile scheduler* til venstre i Figur 6.13 sørger for å tilføre første delen av impulsresponskoeffisientene til øverste konvolusjonsblokk, neste del av koeffisientene til neste konvolusjonsblokk osv. Hver av koeffisientstrømmene null-utfylles til FFT-transformasjonens lengde før overføring. Addisjonskretsen til høyre på figuren utfører sammenkoblingen av delresultatene fra konvolusjonsblokkene til det totale resultatet. Normalt vil en sum av mange delresultater gjøre at sluttresultatet mangedobles i verdi, men siden de forskjellige delresultatene er forskjøvet i tid, riktignok med noe overlapp, så dobles ikke sluttresultatets verdi for hver addisjon. Dette fører til at dataordbredden i addisjonskretsene kan holdes på et moderat nivå. Figur 6.14 viser tidsforløpet i strukturen og sammenligner det mot en enkel konvolusjonsstruktur.



Figur 6.14 Sammenligning av tidsforbruk i en enkel kontra tre parallelle konvolusjonsstrukturer

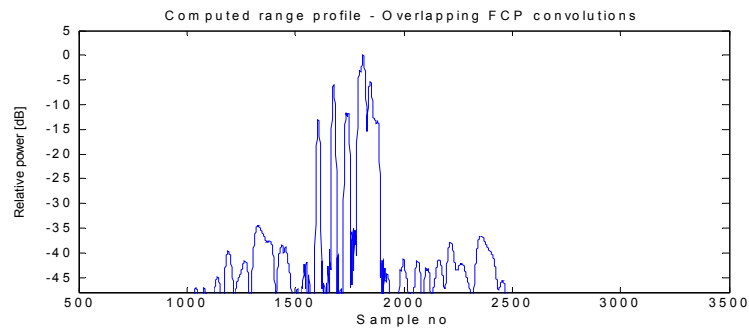
Som Figur 6.14 viser, så kan mye av innskuddsforsinkelsen gjennom konvolusjonsstrukturen reduseres ved hjelp av parallelle grener. Figuren antyder en reduksjon i forsinkelsen på rundt 50%, men eksakt hvor mye som kan spares er avhengig av radarsignalets varighet, avstandsprofilens varighet, hvordan summen av radarsignalets og avstandsprofilens antall sampler går opp i aktuell FFT-størrelse, samt hvor mange parallelle grener det velges å bruke. Jo flere grener, jo mindre antall datapunkter i FFT-transformasjonene, jo hurtigere prosessering. Typisk vil FFT-størrelsen velges noe større enn minimumsverdien, både for at antall grener skal holdes til et moderat antall og for at FFT-størrelsen må være av en gyldig verdi.

Figur 6.14 forutsetter et tidsforbruk i følge ligning (6.1) med konstantene $K=1$ og $M=2$. Det forutsettes også at dataraten inn og ut av konvolusjonsstrukturen er lik prosesseringshastigheten for selve konvolusjonene. Hvis imidlertid dataraten inn og ut av konvolusjonsstrukturen er høyere enn prosesseringshastigheten, noe den ofte er, så må ekstra FIFO-buffere ta opp for hastighetsforskjellen og dette fører til at hastighetsgevinsten ved parallelle grener blir noe mindre enn ellers.

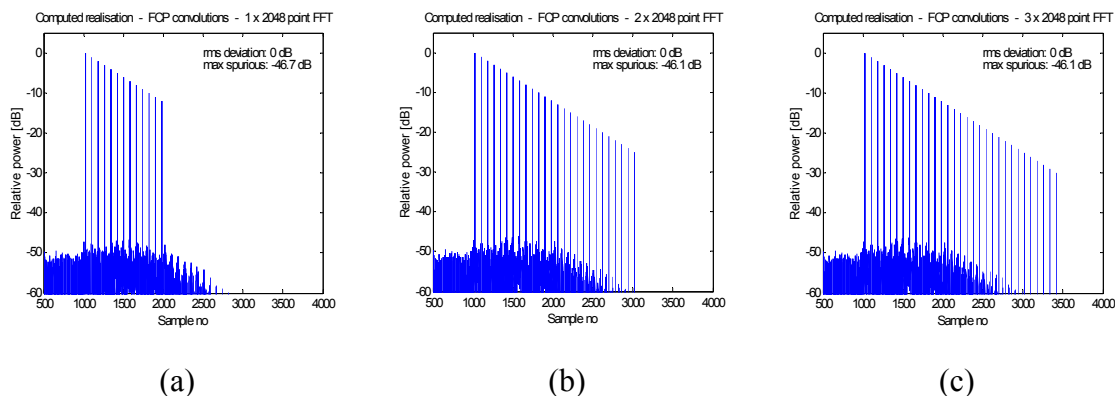
I enkelte tilfeller vil det være nok om deler av det totale avstandsområdet til modulatorene utnyttes, slik at en fullstendig struktur av konvolusjonsblokker kan unngås. Forsinkelsene mellom de enkelte konvolusjonsblokkene må da gjøres programmerbare slik at de enkelte avstandsområdene kan flyttes fritt i avstand. På samme måte må

impulsresponskoeffisientene tilpasses modulatorstrukturen til enhver tid. På denne måten kan noe av kompleksiteten reduseres.

Figur 6.15 viser at strukturen med flere overlappende FCP-konvolusjoner tilsvarer standard FCP-konvolusjon, men vil utføre beregningene vesentlig raskere. Alternativt kan den økte prosesseringshastigheten benyttes til å dekke et større avstandsområde enn det en enkelt FCP-konvolusjon kan klare med et gitt tidsforbruk.



Figur 6.15 Avstandsprofil etter pulskompresjon av et komplekst radarmål generert ved hjelp av 3 stk tidsforskjøvne overlappende FCP-konvolusjoner



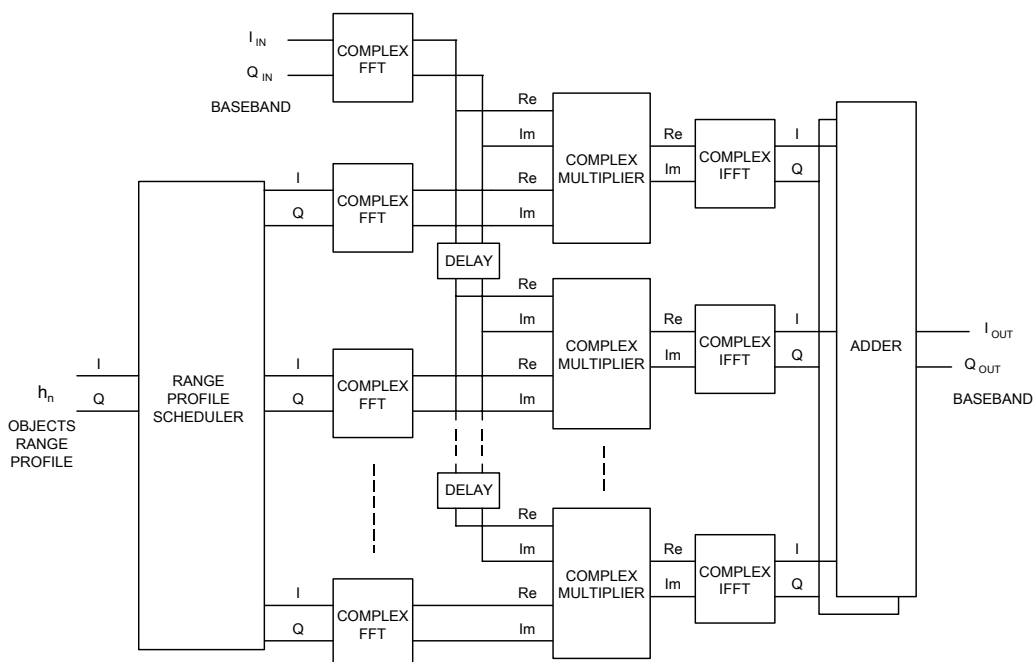
Figur 6.16 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av tidsforskjøvne overlappende FCP-konvolusjoner. Kurve (a) viser avstandsprofilen generert med kun 1 stk 2048 pkt FFT, kurve (b) generert med 2 stk 2048 pkt FFT og kurve (c) generert med 3 stk 2048 pkt FFT.

Figur 6.16 viser at flere parallelle konvolusjoner kan benyttes for å minske innskuddsforsinkelsen eller eventuelt utvide avstandsområdet uten økning i forsinkelsen. Verken rms-avviket eller maksimum spuriøsnivå blir endret i denne omstruktureringen.

6.9 Distribuerte FCP-konvolusjoner

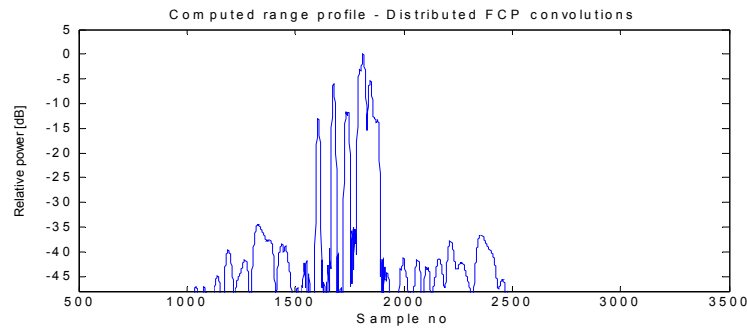
I en struktur med flere FCP-konvolusjoner, vil det være et visst dobbeltarbeid da alle konvolusjonsblokkene beregner FFT-transformasjonen av radarpulsen. Den eneste forskjellen er at radarsignalet er tidsforskjøvet fra konvolusjon til konvolusjon. Egentlig forårsaker disse tidsforsinkelsene i tidsdomenet en konstant fasedreining i frekvensdomenet, ref [11], men siden radarer er ufølsom for en reflektors startfase, så har det ingen innvirkning.

Dermed kan alle transformasjonene forenkles til kun en transformasjon av radarsignalet, og deretter tidsforskyve eller mellomlagre dens frekvensspekter i stedet. Figur 6.17 viser at ved å løse opp den tradisjonelle strukturen kan følgende forenklede og distribuerte strukturskjema fremskaffes:

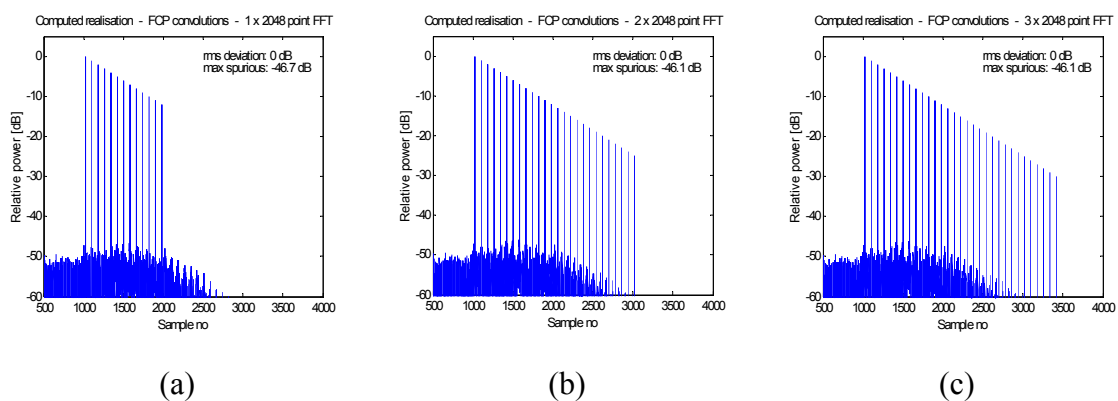


Figur 6.17 Detaljert strukturdiagram for modulator med flere distribuerte FCP-konvolusjoner

Figur 6.18 og Figur 6.19 viser at distribuerte FCP-konvolusjoner gir et identisk resultat i form av rms-avvik og maksimum spuriøsnivå som parallelle FCP-konvolusjoner.



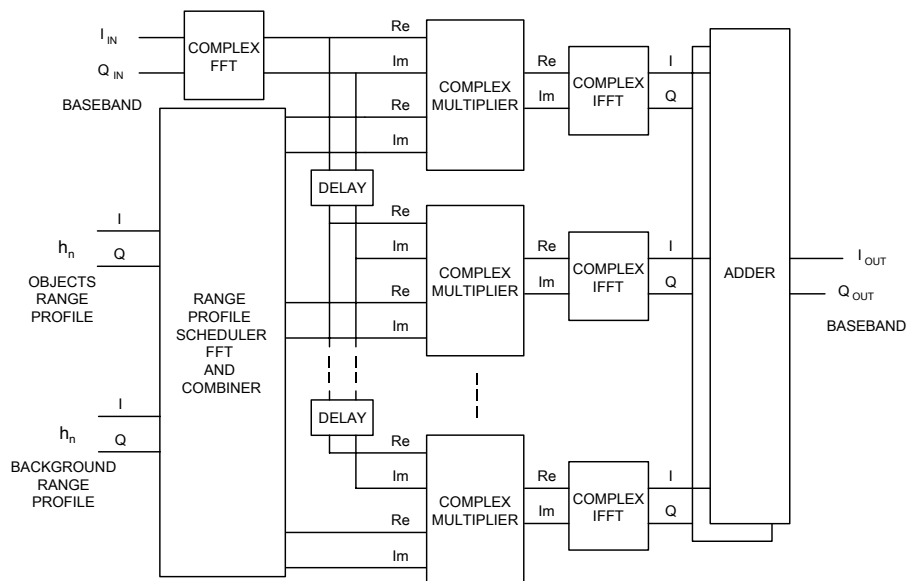
Figur 6.18 Avstandsprofil etter pulskompresjon av et komplekst radarmål generert ved hjelp av 3 stk distribuerte FCP-konvolusjoner



Figur 6.19 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av distribuerte FCP-konvolusjoner. Kurve (a) viser avstandsprofilen generert med kun 1 stk 2048 pkt FFT, kurve (b) generert med 2 stk 2048 pkt FFT og kurve (c) generert med 3 stk 2048 pkt FFT.

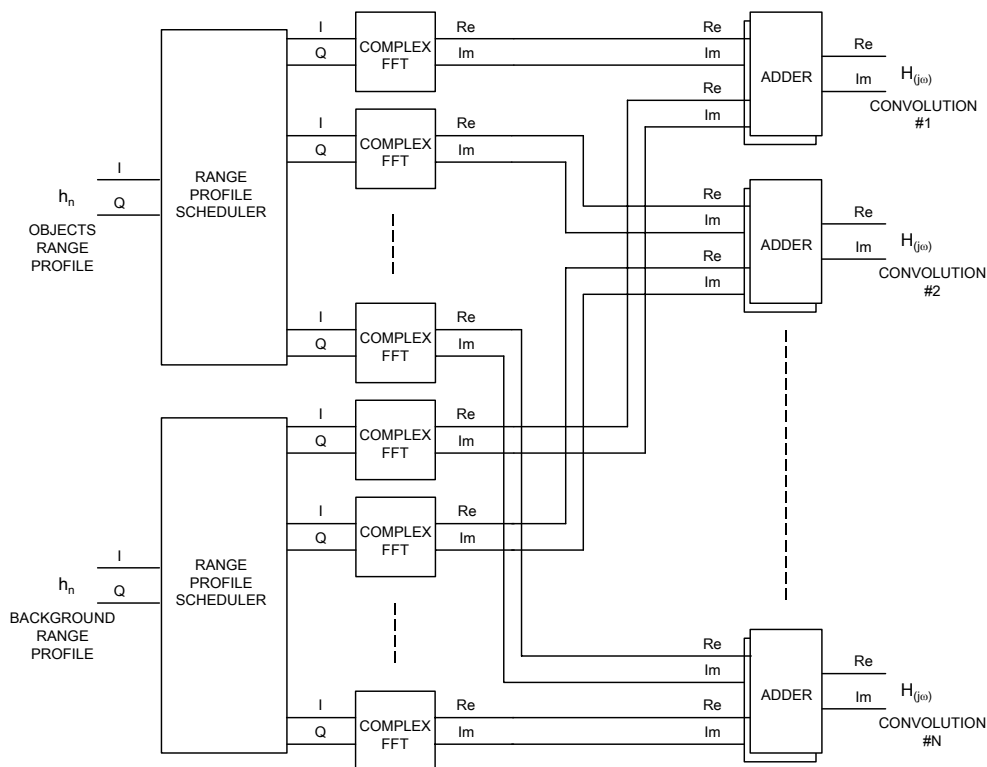
6.10 Distribuerte FCP-konv. med separat modulasjon av objekt og bakgrunn

De samme ønskemål som er beskrevet i kapittel 6.7 om separat modulasjon av objekt og bakgrunn, gjelder for den distribuerte løsningen av FCP-konvolusjon. Figur 6.20 viser strukturdiagrammet:



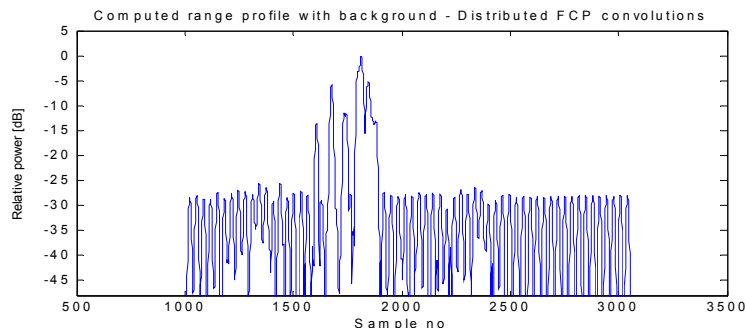
Figur 6.20 Detaljert strukturdiagram for modulator med distribuerte FCP-konvolusjoner med separat modulasjon av objekt og bakgrunn

Som omtalt i kapittel 6.7 blir både objektene og bakgrunnens impulsresponskoeffisienter transformert til frekvensdomenet ved hjelp av FFT-transformasjoner og deretter blir deres spektra addert. Siden denne strukturen gjentas for hver konvolusjon så blir hele oppkoblingen ganske omfattende, ref Figur 6.21:



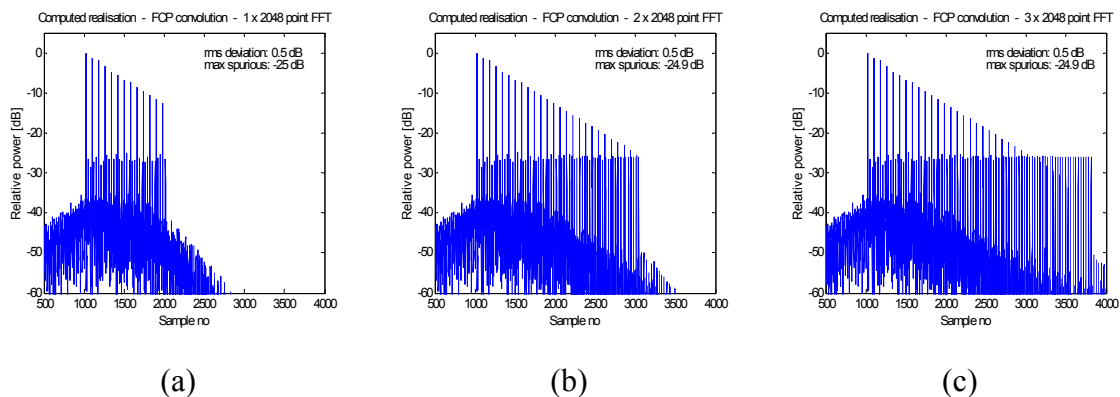
Figur 6.21 Detaljert strukturdiagram for avstandsprofil-fordeler, FFT og kombinerer

Figur 6.22 viser det tidligere eksemplet med avstandsprofil og bakgrunn. Det kan også her ses at bakgrunnens amplitude varierer noe i nærheten av reflektorene. Dette skyldes som sagt pulskompresjonens sideløber av de sterke reflektorene i scenen.



Figur 6.22 Avstandsprofil med bakgrunn etter pulskompresjon av et komplekst radarmål generert ved hjelp av 3 stk distribuerte FCP-konvolusjoner med bakgrunn

Figur 6.23 viser at også distribuerte FCP-konvolusjoner håndterer å generere både objekt og bakgrunn samtidig. Forskjellene i rms-avvik og maksimum spuriøsnivå i forhold til Figur 6.12 er svært små.



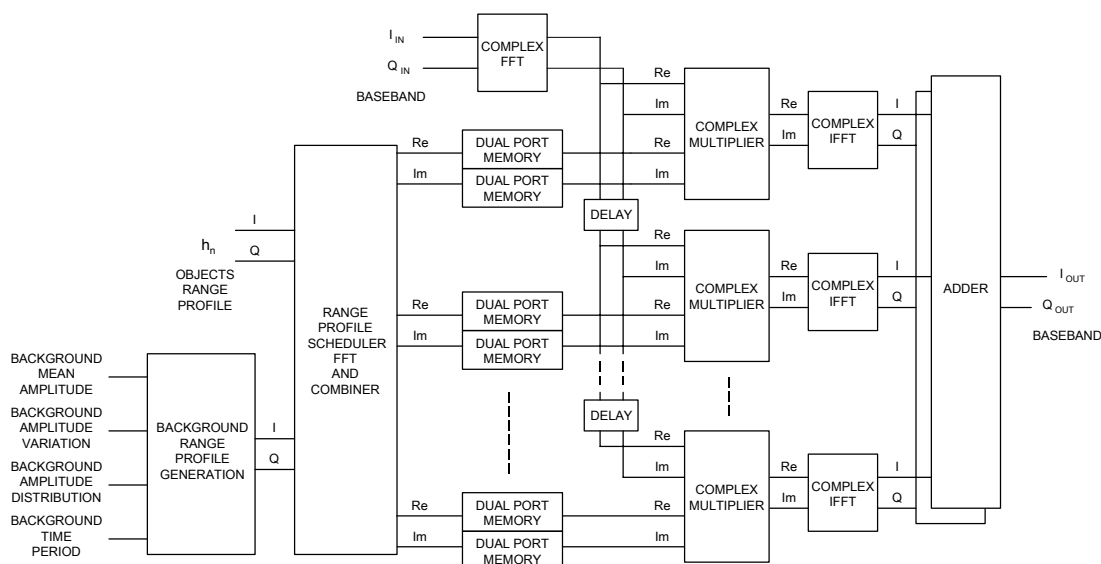
Figur 6.23 Avstandsprofil etter pulskompresjon av referanse 2 generert ved hjelp av distribuerte FCP-konvolusjoner med bakgrunn. Kurve (a) viser avstandsprofilen generert med kun 1 stk 2048 pkt FFT, kurve (b) generert med 2 stk 2048 pkt FFT og kurve (c) generert med 3 stk 2048 pkt FFT.

6.11 Distribuert FCP-konv. med parametrisk generator for bakgrunnsprofil

Dette strukturforslaget er en videreføring av de foregående forslagene, men med to tillegg; parametrisk generering av bakgrunnsprofil og innsetting av databuffere for mellomlagring av

modulasjonsdata mellom høyhastighetslogikk og modulasjonslogikk. De etterfølgende avsnittene omhandler disse tilleggene.

Figur 6.24 viser den komplette struktur med parametrisk generering av bakgrunnsprofilen og strukturen for mellomlagring av både de tilførte impulsresponskoeffisientene h_n for avstandsprofilens objekter og for de automatisk genererte koeffisientene for bakgrunnsprofilen.



Figur 6.24 Detaljert strukturdiagram for modulator med beregnet realisering, inneholdende distribuerte FCP-konvolusjoner og parametrisk generert bakgrunn

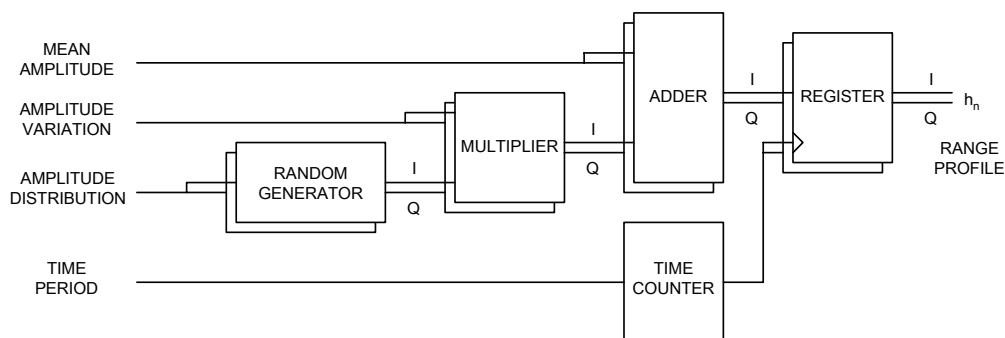
Denne strukturen vil kunne tilpasses de stadig endrede arbeidsforhold, dels ved dens allsidige oppbygning og dels ved å utnytte FPGA-kretsens reprogrammerbare egenskaper.

Strukturen har i stor grad lagt vekt på modulatorens hastighetskrav som er en av de drivende parametere i denne konstruksjonen. Men selv om hastigheten er vektlagt under strukturarbeidet, så er implementasjonen helt avgjørende for at det hele skal nå målet.

6.11.1 Parametrisk generering av bakgrunnsprofil

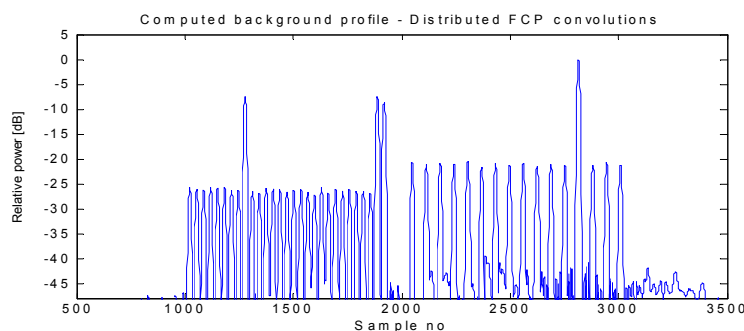
Siden bakgrunnen antakeligvis vil bli generert ved hjelp av et dataprogram med tilfeldige amplituder rundt en midlere verdi, kan kanskje like godt hele bakgrunnsprofilen genereres i modulatorens. Målsettingen er å gi bakgrunnen et troverdig inntrykk.

Som det kan ses av Figur 6.25 kan generatoren stilles inn for midlere amplitude, variasjon i amplituden, variasjonens fordelingsfunksjon og periodetiden for bakgrunnsrefleksjonen. Bakgrunnsprofilen blir her generert i tidsdomenet for senere konvertering til frekvensdomenet ved hjelp av FFT-transformasjoner. I prinsippet kan den samme bakgrunnsprofilen genereres direkte i frekvensdomenet for å slippe FFT-transformasjonene, men siden generatoren da vil inneholde mye kompleks logikk er dette frafalt.



Figur 6.25 Detaljert strukturdiagram av generator for bakgrunnsprofil

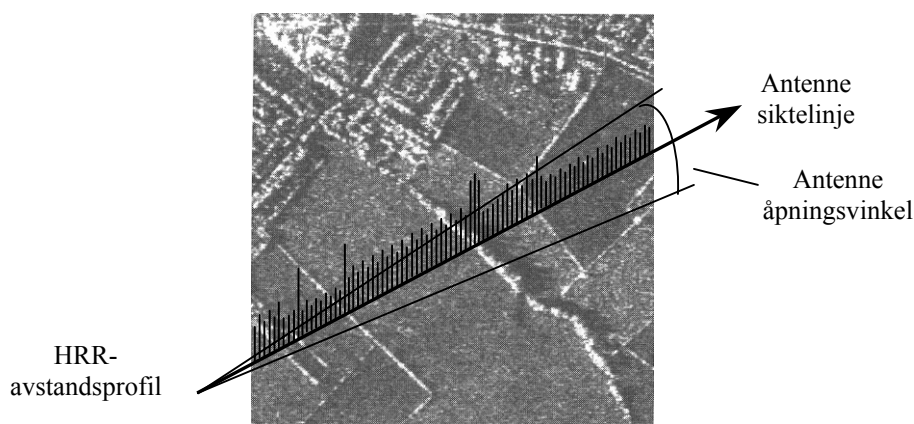
Figur 6.26 viser et enkelt eksempel på bakgrunnsprofil som kan genereres ved hjelp av bakgrunnsgeneratoren. Eksemplet har 'flater' med forskjellig nivå og struktur, og inneholder fire sterke reflektorer hvor to av dem er nært hverandre slik at de danner en reflektor med noe utbredelse. Denne har også en typisk skygge til høyre for seg. Skygger er kjennetegnet av lav refleksjon.



Figur 6.26 Eksempel på scenebakgrunn etter pulskompresjon

Hvis mulighetene i strukturen for bakgrunnsgeneratoren utnyttes fullt ut, vil et nær naturlig utseende bakgrunn kunne genereres. Figur 6.27 viser et høyoppløselig SAR-bilde av et typisk landskap med gress, busker og gjerder, med påtegnet en HRR-avstandsprofil. En HRR-avstandsprofil tilsvarer et snitt gjennom SAR-bildet hvor alle refleksjoner på de enkelte avstandene innenfor radarens antenneåpningsvinkel blir projisert ned på antennens siktelinje og vektorielt summert. Hvert enkelt punkt på avstandsprofilen får dermed en amplitude og fase som er et mål for refleksjonens størrelse ved den enkelte radielle avstand.

Et helt SAR-bilde kan modelleres av en serie HRR-avstandsprofiler som er generert over en gitt tid med individuell modulasjon. En SAR-radar benytter dopplerprosessering i tillegg til avstandsprofilene for å fastslå hvor i 2D-scenen en refleksjon kommer fra.



Figur 6.27 SAR-bilde av et landskapsområde, ill fra [10], med eksempel på en typisk HRR-avstandsprofil inntegnet

På grunn av at generatoren automatisk genererer bakgrunnen ved hjelp av noen få parametere, vil modulasjonsdatamengden reduseres drastisk. Typisk vil kun 4 parameterverdier overføres for hver gang bakgrunnen endrer struktur eller utseende.

6.11.2 Databuffere mellom høyhastighetslogikk og modulasjonslogikk

Avhengig av radartype og ønskemålet med jammingen vil det være forskjellige krav til oppdatering av modulasjonsdata. Mens det ved en type radar er tilstrekkelig med de samme modulasjonskoeffisientene gjennom flere titalls radarpulser, vil det ved andre radartyper være påkrevd med nye koeffisienter for hver radarpuls. De svært varierende situasjonene fordrer en fleksibel løsning.

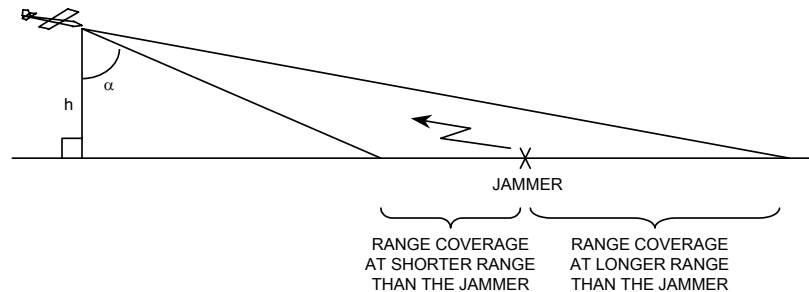
En metode hvor to-port buffere benyttes som mellomagre mellom selve konvolusjonslogikken og avstandsprofilenes modulasjonslogikk vil kunne håndtere begge situasjonene beskrevet ovenfor. To-port bufferne vil bli lest for hver konvolusjonsberegning, mens oppdateringen av bufferne er gitt av modulasjonslogikken. Om modulasjonslogikken oppdaterer bufferne for hver jammeutsending eller f.eks for kun hver 20 utsending er gitt av modulasjonslogikkens interne styring og vedkommer ikke selve konvolusjonene.

6.12 Eksempel I på valg av realiseringsstruktur

I disse regneeksemplene skal det gjennomgås hvilke problemstillinger som vil være relevante ved valg av struktur for realisering av modulatoren. Først og fremst må alle relevante radarparametere og den geometri som modulatoren skal brukes under fastsettes. Av radarparametere må hovedsaklig bølgeformens varighet og dens repetisjonsintervall kartlegges. Av geometriske data er ønsket avstandsdekning og eventuelt sensorens høyde og dens siktevinkel ned på bakken.

Dette eksemplet tar spesielt for seg kampfly bakkeavbildning med HRR-radar, hvis geometri kan ses i Figur 6.28. Radaren ombord i flyet har avstandsoppløsning lik 1 meter og dens utsendte bølgeform har derav en båndbredde lik 150 MHz. Radaren måler som kjent

tidsforsinkelser til refleksjonspunkter på bakken og gjengir de på en radarskerm med angivelse i nautiske mil eller meter. Denne avstanden er ikke målt avstand på bakken, men avstanden i radarens pekeretning – radiell avstand. Hvis dette skal regnes om til avstand på bakken, må flyets høyde og radarens pekevinkel tas med i beregningene.



Figur 6.28 Geometri ved kampfly bakkeavbildning

I Figur 6.28 kan jammerens posisjon ses innenfor flyradarens operative område. Falske mål som skal plasseres på lengre avstand fra radaren enn jammerens posisjon må forsinkes i jammeren for å illudere en større avstand. I tillegg må falske mål som ønskes plassert på kortere avstand enn jammeren forsinkes en hel PRI minus ønsket avstand fra jammeren. Da vil radarsignalet som mottas i jammeren først bli brukt som jamming i den etterfølgende PRI.

Det antas her at ved HRR bakkeavbildning fra et kampfly er radarens PRF i området 1 kHz dvs med PRI rundt 1 ms, og bølgeformens varighet er i området 10 μ s. På grunn av radarens hurtige modusskift er det krav til at forsinkelsen gjennom modulatorens må holdes på et absolutt minimum. Hvor lang kan modulatorens avstandsområde bli når det forutsettes at radarsignalet samples med 500 MSPS og tilgjengelig prosesseringshastighet er 125 MHz?

Følgende realiseringsstrategier blir gjennomgått:

6.12.1 Standard konvolusjon

Det store antall refleksjoner over et større avstandsområde som her skal modelleres gjør at standard FIR-filter realisering kan utelukkes helt. Kompleksiteten ved en slik realisering vil overgå alle tilgjengelige ressurser.

6.12.2 FCP-konvolusjon

Med en enkel FCP-konvolusjon kan typisk en femtedel av tiden tilgjengelig benyttes til hver av de fem prosesseringsfasene når dataraten inn og ut av konvolusjonen er høyere enn prosesseringshastigheten for selve konvolusjonen, ref Figur 6.14 og tilhørende tekst. I dette tilfellet er dataraten inn og ut på 500 MSPS, mens prosesseringshastigheten er 125 MHz. Siden et av kravene for konstruksjonen var minimum forsinkelse, så legges det opp til at minimumsforsinkelsen ikke skal overstige en PRI.

Dette fører til at $1 \text{ ms} / 5 = 200 \text{ } \mu\text{s}$ vil være tilgjengelig for hver av de fem prosesseringsfasene. Siden radarens bølgeform har varighet som er vesentlig mindre enn tilgjengelig prosesseringsstid så kan beregningene utføres uten problem innenfor en PRI.

En vanlig FFT-transform kan typisk prosessere et datapunkt pr klokkesyklus ved 125 MHz rate. I løpet av $200 \text{ } \mu\text{s}$ kan dermed teoretisk maksimalt 25000 datapunkter prosesseres. Med de gjeldende størrelsene for FFT er $2^{14} = 16384$ datapunkter den nærmeste størrelsen under den teoretisk maksimale verdien. Samplingen av radarens bølgeform opptar $10 \text{ } \mu\text{s} * 500 \text{ MSPS} = 5000$ datapunkter. Antall datapunkter for modulatorens avstandsområde blir dermed $16384 - 5000 = 11384$ datapunkter som omregnet blir $22,768 \text{ } \mu\text{s}$ forsinkelse eller totalt avstandsområde på $3,42 \text{ km}$ radiell avstand.

For å øke avstandsområdet kan en eller flere av følgende gjøres; innføre hurtigere prosessering dvs prosessering ved høyere klokke rate enn 125 MHz, senke kravet til maksimalforsinkelsen og / eller benytte parallelle konvolusjonsstrukturer.

6.12.3 FCP-konvolusjon med separat modellering av objekt og bakgrunn

Denne realiseringen er aktuell siden modellering av bakkeavbildende radar medfører store mengder bakgrunn som skal modelleres og som ofte er uavhengig av objektene. Det avstandsområdet som denne strukturen kan modellere er ellers lik det vanlig FCP-konvolusjon kan gjøre.

I denne strukturen er det som i de andre strukturene en stor fordel om modulasjonsdata for objektene og / eller bakgrunnen kan transformeres til frekvensdomenet på forhånd slik at mange tidskritiske beregninger kan unngås i sanntid.

6.12.4 Parallele / distribuerte FCP-konvolusjoner

Parallele / distribuerte FCP-konvolusjoner er i prinsippet den samme strukturen, men den distribuerte varianten har færre beregninger for ellers likt resultat.

Fordelen med parallelle / distribuerte konvolusjoner er at avstandsområdet deles mellom flere mindre konvolusjoner som er betydelig raskere å beregne og som dermed gir mindre innskuddsforsinkelse.

Det foreslås nå å utvide strukturen beregnet i avsnitt 6.12.2 slik at den kan dekke et større avstandsområde. Opprinnelig var avstandsområdet på $3,42 \text{ km}$ radiell avstand. Hvis et radielt avstandsområde på 20 km skal dekkes trengs $20 \text{ km} / 3,42 \text{ km} = 6$ parallelle konvolusjonsgrener.

6.12.5 Distribuert FCP-konv. med separat modulasjon, evt. med parametrisk generator

Disse realiseringsforslagene er av samme grunn som forslag beskrevet i avsnitt 6.12.3 aktuelle siden modellering av bakkeavbildende radar medfører store mengder bakgrunn som skal modelleres og som ofte er uavhengig av objektene. Det avstandsområdet som disse strukturene kan modellere er ellers lik det parallelle distribuerte FCP-konvolusjoner kan gjøre.

6.12.6 Sammendrag av realiseringsforslag for eksempel I

Tabell 6.1 viser en enkel sammenligning av realiseringsforslagene for eksempel I.

Struktur	Forsinkelse	Avstandsområde	Kompleksitet
Std konvolusjon	Svært liten	Lite	Svært stor
FCP-konvolusjon	1 PRI	3,42 km	Liten
FCP-konvolusjon m/separat modellering objekt / bakgrunn	1 PRI	3,42 km	Liten
Parallelle/distribuerte FCP-konvolusjoner	1 PRI	20,52 km	Middels
Parallelle/distribuerte FCP-konvolusjoner m/separat modellering objekt / bakgrunn	1 PRI	20,52 km	Middels / stor

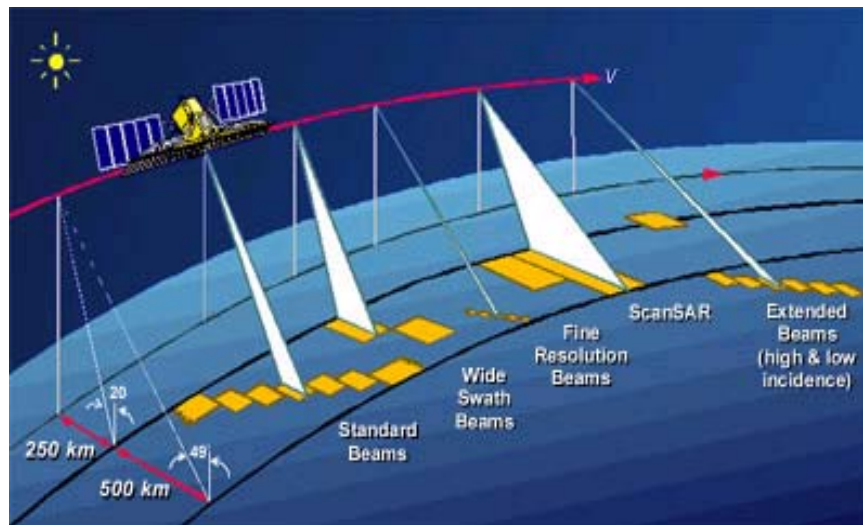
Tabell 6.1 Oppsummering av nøkkelparametere for realiseringsforslagene i eksempel I

6.13 Eksempel II på valg av realiseringsstruktur

Dette eksemplet skal på samme måte som forrige eksempel gjennomgå relevante problemstillinger ved valg av realiseringsstruktur for modulatorene.

Eksempel II tar for seg satellitt bakkeavbildning med geometri som vist i Figur 6.29. Regneeksemplet benytter radarparametere fra den kanadiske Radarsat-1 SAR-radaren. Denne radaren har en rekke operasjonsmodi, men i dette eksemplet skal parametere for standard mode benyttes, ref Tabell 6.2. Radarjammeren forutsettes å være plassert på bakken

innenfor SAR-radarens antennelobe.



Figur 6.29 Geometri for Radarsat-1 SAR bakkeavbildning, ill fra [24]

Oppløsning	25 x 28 m
Sporbredde	100 km
Innfallsvinkel	20° – 49°
Banehøyde	798 km polarbane
Omløpstid	100,7 min
Pulsvarighet	42 μ s
PRF	1270 Hz – 1390 Hz
Frekvensbånd	C-bånd – 5,3 GHz

Tabell 6.2 Karakteristiske parametere for Radarsat-1, standard mode

Den store forskjellen fra forrige eksempel er det store avstandsområdet – 100 km som ønskes dekket i dette tilfellet, samt at bølgeformens varighet er på hele 42 μ s. Det bør bemerkes at i mange tilfeller kan troverdig jamming utføres selv om ikke hele radarens avstandsområde dekkes.

Forsinkelsen gjennom modulatoren er ikke fullt så kritisk her som ved kampfly bakkeavbildning, men den bør begrenses til det nødvendigste. Radarsignalet samples med 500 MSPS som før, mens det nå benyttes 'state of the art' pipeline FFT signalprosessering med 400 MHz klokke-rate og opptil $2^{18} = 262144$ datapunkter. Hvor mange parallelle FCP-konvolusjoner trengs for å løse oppgaven?

Med 1 PRI som maksimal forsinkelse gjennom modulatorene er $1/1390 \text{ Hz} = 720 \mu\text{s}$ tilgjengelig. Ved å benytte ligning (6.2) kan maksimalt antall datapunkter i FCP-konvolusjonen finnes. Av de standardiserte størrelsene er 65536 datapunkter den største som benytter mindre tid enn $720 \mu\text{s}$. Den neste størrelsen, 131072 datapunkter, bruker $735 \mu\text{s}$ som er litt for lang tid innenfor en PRI.

Samplingen av radarens bølgeform opptar $42 \mu\text{s} * 500 \text{ MSPS} = 21000$ datapunkter og antall datapunkter for modulatorens avstandsområde blir dermed $65536 - 21000 = 44536$. Omregnet til radiell avstand blir dette $13,36 \text{ km}$, som er operativt relevant i høyeste grad!

Med radarens lange bølgeform, dekning i avstand og oppløsning vil ca 250 overlappende bølgeformer genereres som jamming hvis alle radarens oppløsningsceller skal dekkes. Dette stiller høye krav til dynamikk og fordrer riktig skalering av reflektorene slik at metning av addisjonsnettverk og D/A-konverterne unngås.

For å dekke det ønskede avstandsområdet på 100 km , med minste innfallsvinkel, vil totalt $100 \text{ km} * \cos 20^\circ / 13,36 \text{ km} = 8$ parallelle konvolusjonsgrener være nødvendig. Denne strukturen vil bli meget kompleks siden mange parallelle grener med store høyhastighetskonvolusjoner inngår. Om strukturen skal innbefatte separat modellering av objektene og bakgrunnen vil kompleksiteten ytterligere vokse.

Hvis SAR-radaren hadde blitt operert i en mode hvor maksimal PRF var mindre enn 1360 Hz , ville en enkelt FCP-konvolusjon med 131072 datapunkter dekke $34,7 \text{ km}$ på bakken med laveste innfallsvinkel. Og hvis i tillegg den maksimale innfallsvinkelen på 49° hadde blitt benyttet, ville $46,0 \text{ km}$ på bakken blitt dekket.

Hvis ønsket om kun 1 PRI forsinkelse gjennom modulatorene endres til 3 PRI, vil følgende resultat fremstå:

Tilgjengelig tid for konvolusjonsstrukturen: $3 * 720 \mu\text{s} = 2160 \mu\text{s}$

FCP-konvolusjon med 262144 datapunkter beregnes i løpet av: $1471 \mu\text{s}$ ref ligning (6.2).

Modulatorens avstandsområde blir dermed: $76,06 \text{ km}$ med den minste innfallsvinkelen.

Totalt trengs det nå 2 parallelle konvolusjonsgrener for å dekke et 100 km avstandsområde. Dette må anses å være av en realiserbar kompleksitet, selv om hver av FFT-transformene er store og komplekse høyhastighetstransformasjoner.

Tabell 6.3 gir en oppsummering av realiseringsforslagene.

Struktur	Forsinkelse	Avstandsområde	Kompleksitet
Parallele/distribuerte FCP-konvolusjon	1 PRI	100 km	Meget stor
Parallele/distribuerte FCP-konvolusjoner m/separat modellering objekt / bakgrunn	1 PRI	100 km	Svært stor
Parallele/distribuerte FCP-konvolusjon	3 PRI	100 km	Stor
Parallele/distribuerte FCP-konvolusjoner m/separat modellering objekt / bakgrunn	3 PRI	100 km	Stor / meget stor

Tabell 6.3 Oppsummering av nøkkelparametere for realiseringsforslagene i eksempel II

6.14 Diskusjon

Dette kapitlet diskuterer analysens enkelte løsningsforslag for beregnet realisering av digital modulator. Hvert forslag blir omhandlet i eget avsnitt.

6.14.1 Standard konvolusjon

Standard konvolusjon eller et standard FIR-filter er en lett tilgjengelig metode for modellering av radarrefleksjoner. Et standard FIR-filter vil ha seksjoner som er avskilt fra hverandre med en fast forsinkelse gitt av samplerraten. En løsning bygget opp på denne måten har et betydelig problem når store antall refleksjoner over et større avstandsområde skal modelleres. I utgangspunktet vil FIR-filteret bestå av en kompleks multiplikasjon for hver forsinkelsesavtapning. Med 500 MSPS datarate trengs det 3333 komplekse multiplikasjoner pr km avstandsområde. Antallet multiplikasjoner er kanskje ikke det største problemet, men at alle multiplikasjonene må regnes for hvert eneste sample av radarsignalet gjør dette til en meget stor beregningsbyrde.

Selv et FIR-filter hvor forsinkelsene er programmerbare, med andre ord direkte realisering, vil bli kompleks hvis antall refleksjoner blir stor. Standard konvolusjon har sin styrke ved svært liten innskuddsforsinkelse, men kan i praksis bare benyttes ved modellering av enkelte objekter med liten utstrekning.

6.14.2 FCP-konvolusjon

FCP er en effektiv implementasjon av konvolusjon basert på FFT-implementasjonen av DFT. Selve beregningen foregår her i frekvensdomenet, da tidsdomenets konvolusjon blir til multiplikasjon i frekvensdomenet. FCP har vesentlig større ytelse i hastighet enn standard konvolusjon, og ytelsesforskjellen bare vokser etter hvert som antall datapunkter i inngangssekvensene øker.

En ulempe i forbindelse med modellering av radarrefleksjoner med konvolusjon basert på FFT er at antall datapunkter i FFT-beregningene må minst være lik summen av antall datapunkter i de to tidsseriene som skal beregnes. Ved lange pulskomprimerte bølgeformer blir dermed antall datapunkter i FFT-beregningene store og beregningstiden øker tilsvarende. Så lenge tidskravene ikke er for harde, vil FCP-konvolusjon være et godt valg. Strukturens kompleksitet er relativt liten med 3 komplekse FFT-transformasjoner og 1 kompleks multiplikasjon.

6.14.3 FCP-konvolusjon med separat modellering av objekt og bakgrunn

Denne strukturen har mulighet for separat håndtering av modellerte objekter og bakgrunn. I mange tilfeller vil det være ønskelig å generere modulasjonsdata for objektet og dens forflytning og rotasjon i en prosess avskilt fra den mer rolige bakgrunnen med færre distinkte reflektorer, men med mer vekt på de riktige statistiske fordelingene.

I kompleksitet er denne strukturen noe mer omfattende enn vanlig FCP-konvolusjon, med 1 kompleks FFT og 2 reelle addisjoner i tillegg, men hastighetsmessig er det nærmest ingen forskjell. Dette taler til fordel for strukturen med separat modulasjon.

6.14.4 Parallelle FCP-konvolusjoner

Ved bruk av et antall parallelle konvolusjonskretser kan beregningshastigheten i mange tilfeller økes betraktelig. Minimumsforsinkelsen gjennom modulatorene er imidlertid begrenset av antall datapunkter i FFT-beregningene, og de må minst være lik summen av antall datapunkter i radarens bølgeform og i aktuelt avstandsområde som skal dekkes. Når flere parallelle kretser benyttes blir imidlertid det aktuelle avstandsområdet delt på antall parallelle grener. Strukturens kompleksitet vokser lineært med antall grener og må sies å være relativt kompleks. Økningen i kompleksitet fra en enkelt FCP-konvolusjon til en parallell FCP-konvolusjon med to grener er på mer enn en dobling.

Først og fremst er denne metoden egnet til å øke modulatorens avstandsområde uten at beregningstiden øker. Dersom radarens bølgeform består av relativt få datapunkter vil denne strukturen også kunne gi en meget hurtig beregningshastighet, selv om aktuelt avstandsområde er stort.

6.14.5 Distribuerte FCP-konvolusjoner

Denne strukturen er i prinsippet lik strukturen med parallelle konvolusjoner hvor en del av beregningene ble utført flere ganger. Dette er forenklet i den distribuerte konvolusjonsstrukturen og kompleksiteten vokser ikke lenger lineært med antall grener. Ytelsesmessig er strukturen identisk med parallelle konvolusjoner, og må fortsatt sies å være relativt kompleks av natur.

6.14.6 Distribuert FCP-konvolusjon med separat modulasjon av objekt og bakgrunn

Distribuert FCP-konvolusjon hvor modulasjonsdata tilføres separat for objektene og for bakgrunnen er en kombinasjon av strukturforslagene presentert i kapitlene 6.7 og 6.9. Kompleksiteten i denne strukturen er stor, først og fremst på grunn av de mange FFT-transformasjonene som trengs når antall grener øker og det skal være separat modulasjon av både objekt og bakgrunn. Hastighetsmessig er strukturen nær identisk med strukturen for distribuerte FCP-konvolusjoner.

6.14.7 Distribuert FCP-konvolusjon med parametrisk generator for bakgrunnsprofil

Denne strukturen bringer modulatoren opp på et høyere ambisjonsnivå enn en vanlig modulator. Her kan parametere lastes inn for hvordan bakgrunnens utseende i en avstandsprofil skal fremstå. Parametersettet vil normalt bli endret flere ganger i løpet av genereringen av en avstandsprofil dvs i løpet av pulsrepetisjonsintervallet, men bare når bakgrunnen skal endre utseende. Dette fører til en slags høynivå beskrivelse av bakgrunnen.

Denne strukturen har også innlagt et sett med buffere for mellomlagring av modulasjonsdata for hver gren med konvolusjoner. På denne måten er strukturen langt bedre rustet til å håndtere varierende krav til oppdatering av modulasjonsdata, siden de kun trenger å overføres ved skifte i modulasjonen. Kravet til overføringskapasitet av modulasjonssignalene blir på denne måten redusert.

Kompleksiteten blir i denne strukturen stor på grunn av mye høyhastighetslogikk og mye ekstra funksjonalitet. Hvis alt ses under ett, så kan likevel den økte kompleksiteten forsvares ut i fra den ekstra ytelsen.

6.15 Konklusjon

Denne analysen viser at flere strukturer for beregnet realisering av en digital modulator kan være aktuelt. Strukturforslaget - *Distribuert FCP-konvolusjon med parametrisk generator for bakgrunnsprofil* er forslaget med høyest ambisjonsnivå samtidig som det er best tilpasset forskjellige situasjoner.

I en del situasjoner vil enkelte av de mindre ambisiøse strukturforslagene kunne være en bedre løsning, da de mer komplekse strukturene kan vise seg vanskelig å realisere innenfor gjeldene teknologiske begrensninger.

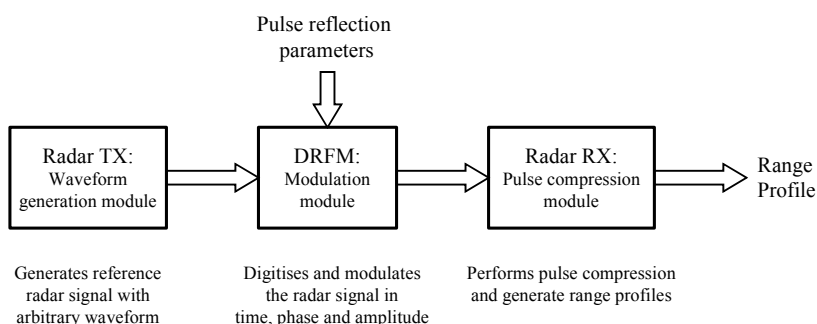
Ved å utnytte FPGA-kretsens fleksibilitet kan modulatorens struktur omprogrammeres fra en situasjon til en annen, og på den måten er det ikke så viktig at en spesiell generell struktur utvikles for alle tenkelige situasjoner.

7. Analyse av digitaliseringen i en DRFM brukt mot HRR-radar

7.1 Innledning

En avgjørende parameter i en DRFM-konstruksjon er antall bit som radarsignalene skal representeres med internt i systemet. Når DRFM benyttes mot høyoppløselig radar vil kompleksiteten i signalene øke etter hvert som moduleringen utføres, og da er det rimelig å anta at modulasjonsberegningene og D/A-konverterne vil trenge flere bit oppløsning enn A/D-konverterne og lagringen av innkommende radarsignal.

Analysen baserer seg på datamodellering av radarens pulskoding og pulskompresjonsalgoritmer, og på DRFM digitalisering og modulering av jammesignalene. Figur 7.1 viser en skjematisk beskrivelse av datamodellen som ble brukt.



Figur 7.1 Skjematisk beskrivelse av datamodellen

7.2 Hva digitaliseringen gjør med signalene

Under digitaliseringen skjer en endring av signalene – de analoge signalene begrenses i amplitude til et fåtall diskrete lovlige verdier. Jo flere bit som benyttes i kvantiseringen jo mindre informasjon går tapt.

Den feilen som oppstår under digitaliseringen kalles digitaliseringsfeil eller kvantiseringsfeil e_m , og kan maksimalt bli $q/2$, hvor q er verdien på det minst signifikante bit i valgte kode. Dette forutsetter imidlertid at A/D-konverteren alltid velger den optimale digitale verdi.

Digitaliseringsfeilen e_m gir opphav til et nærmest hvitt frekvensspektrum i området fra null til samplingsfrekvensen, ref [11] og det kan vises at:

Midlere verdi for $e_m = 0$

$$\text{rms-verdi for } e_m = \frac{q}{\sqrt{12}} \quad (7.1)$$

Maksimal verdi for $e_m = q/2$

hvor q er verdien på det minst signifikante bit og rms-verdi står for kvadratroten av midlere kvadratisk avvik.

Men, når et signal er sammensatt av et stort antall digitaliserte signaler med forskjellig frekvens, amplitude og fase vil frekvensspekteret ikke bli flatt. Det vil oppstå spuriøse komponenter langt over det midlere nivå på grunn av konstruktiv interferens mellom enkelte av signalkomponentene. Det kan antas at jo flere enkeltsignaler jo mer konstant blir det midlere spuriøsnivå relativt sett.

7.3 Kompleksiteten i generert jammesignal mot høyoppløselig radar

I en DRFM konstruert for bruk mot en tradisjonell radar vil vanligvis antall bit i A/D- og D/A-konverterne være lik. Datahukommelsen som lagrer den samlede radarpulsen vil også ha samme bitbredde. Gjengivelse av en enkelt reflektor om gangen vil ikke kreve mer, men antall bit i digitaliseringen vil påvirke gjengivelsen av reflektoren og den medfølgende digitaliseringsstøyen.

Men, en DRFM som er konstruert for å benyttes mot en høyoppløselig radar trenger større dynamikk i utsendt jammesignal enn i mottatt radarpuls. Dette fordrer dermed et større antall bit i dataprosesseringen etter lagring og i D/A-konverterne. Kravet til stor dynamikk skyldes det store antall overlagrede refleksjoner som må gjengis med høy nøyaktighet for å generere et troverdig falskt mål for en høyoppløselig radar. Så lenge refleksjonene er overlagret hverandre vil jammesignalet være lik vektorsummen av de enkelte refleksjonene. Maksimalt vil jammesignalet oppta et dataord med følgende bitbredde i tillegg til opprinnelig bitbredde for en enkelt refleksjon:

$$\text{Økning i bitbredde} = \text{ceil}(\log_2 N) \quad (7.2)$$

hvor bitbredde er antall bit i D/A-konverterne og N er totalt antall reflektorer.

Normalt vil ikke alle reflektorene ha like stor amplitude og dermed er det ikke påkrevd med en fullt så stor økning i antall bit. F.eks kan det antas at de forskjellige reflektorenes amplituder fordeles med uniform fordeling over et gitt dynamisk område. Dermed blir kravet til antall bit mindre.

Det er heller ikke realistisk at alle reflektorene skal oppnå konstruktiv interferens samtidig, i hvert fall ikke ut over et meget kort tidsrom. Sannsynligheten for interferens mellom alle reflektorene blir mindre og mindre etter hvert som antall reflektorer øker, men samtidig øker sannsynligheten for interferens mellom enkelte reflektorer. I alle fall er det viktig å ta stilling

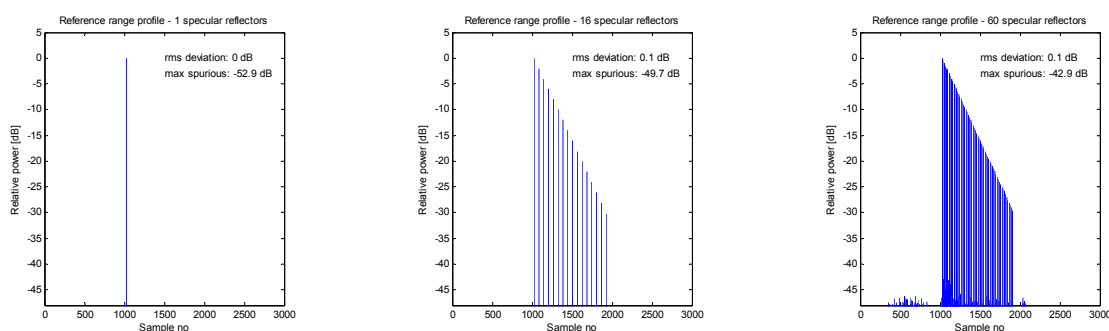
til dette når store antall reflektorer skal behandles, da potensialet for å spare på antall bit er størst her.

7.4 Definisjon av referansesignal

For at referansesignalene skal være realistiske signal er de basert på en chirp-puls som er blitt modulert av refleksjonspunkter fra et tenkt objekt. Objektene har ingen naturlig form, men deres reflektorer er i posisjon og amplitude valgt slik at referansesignalene skal ha egenskaper som avslører avvik på en lett synlig og målbar måte. Chirp-pulsen har et tid-båndbredde produkt på 1023 og har etter pulskompresjon i radarmottakeren et lavt sidelobenivå. For at chirp-pulsens sidelobenivå ikke skal påvirke referansesignalene er det viktig at den valgte pulsmodulasjon har vesentlig lavere sidelobenivå enn digitaliseringsstøyen fra A/D-konverterne og generert spuriøsnivå fra addisjonen av de enkelte reflektorene.

Til analysen er det laget 3 referansesignal som er modulert av henholdsvis 1, 16 og 60 reflektorer. Referansesignal 1 består som sagt av 1 reflektor med amplitude på 0 dB. Referansesignal 2 har 16 reflektorer hvor den første reflektoren med amplitude 0 dB og de etterfølgende med en fast synkende amplitude på 2 dB fra reflektor til reflektor, noe som gir 30 dB total dynamikk. Avstanden fra reflektor til reflektor er fast på 60 sample. Referansesignal 3 har mye til felles med referansesignal 2, men består totalt av 60 refleksjonspunkter med en avstand på 15 sample. Amplitudeforskjellen fra reflektor til reflektor er her 0.5 dB.

Referansesignalene er vist i Figur 7.2 etter pulskomprimering i datamodellen. Referansesignalene er prosessert med svært høy oppløsning og kan dermed betraktes som om de var 'analoge signaler'. Figuren viser også beregnet maksimal rms-avvik og maksimal spuriøsnivå, som er gitt av referansesignalenes bølgeform og pulskompresjonens undertrykking av sidelober.



Figur 7.2 Referansesignal 1, 2 og 3 etter kompresjon i datamodell

7.5 Mål for signalenes 'godhet' etter digitalisering, modulering og komprimering

Som tidligere nevnt benyttes datamodellering i denne analysen, se Figur 7.1. En radarsendermodell genererer en radarpuls med pulskoden chirp. Deretter digitaliseres radarpulsen med et ønsket antall bit og moduleres i tid og amplitude av modulormodellen. Til slutt foretar radarmottakermodellen pulskompresjon av den kodede modulerte radarpulsen og en avstandsprofil blir generert.

På denne måten utføres digitaliseringen og moduleringen av radarsignalene som i en DRFM og resultatet sjekkes etter pulskompresjonen som i en høyoppløselig radar. Som mål for signalenes godhet blir det beregnet rms-avvik og det høyeste spuriøsnivå.

7.6 Gjennomføringen av analysen

For å skille effektene som oppstår i A/D-konverterne fra effektene i modulator og D/A-konverterne, er det valgt å fokusere på en ting av gangen. Første del av analysen er derfor viet A/D-konverterne for I og Q-basebåndsignalene. Dette gjøres ved at antall bit i A/D-konverterne varieres mens antall bit i modulator og D/A-konverterne holdes konstant høy. Her er målet å finne ut hvordan antall bit i A/D-konverterne påvirker pulskompresjonsprosessen i radarmottakeren.

Andre del av analysen tar for seg modulatorene. Modulatoren består av tre hoveddeler; modulasjon av tidsforsinkelse, modulasjon av amplitude og til slutt summering av alle de modulerte signalene. Siden denne analysen er basert på EKKO II arkitekturen, ref kapittel 4.6, er oppløsningen i tidsforsinkelsene gitt av modellens samplerate og inngår i alle beregninger og kurver som blir presentert. Eventuelle avvik vil gjenspeiles i rms-avvikene som beregnes.

Da en DRFM for høyoppløselige radarer trenger stor amplitudedynamikk i modulatorene, så fokuseres det nærmere på egenskapene til å håndtere både sterke og svake reflektorer. Til dette benyttes referansesignal 2 som inneholder et antall reflektorer med forskjellige amplituder. Målet er å finne ut hvordan antall bit i modulatorens amplitudemodulasjon påvirker gjengivelsen av reflektorenes amplitudenøyaktighet.

Modulatoren summerer tilslutt alle bidragene som er modulert i tid og amplitude. Pga det store antall reflektorer som skal adderes, er modulatorens evne til å håndtere mange samtidige reflektorer av interesse. Til dette benyttes referansesignal 3 med hele 60 reflektorer, hvor hver av dem har forskjellig amplitude. Denne del av analysen tar også for seg D/A-konverterne, da de naturlig får samme bitbredde som addisjonskretsens utgangssignal. Målet er å finne ut hvordan antall bit i addisjonsnettverket og i D/A-konverterne påvirker gjengivelsen av et stort antall samtidige signaler.

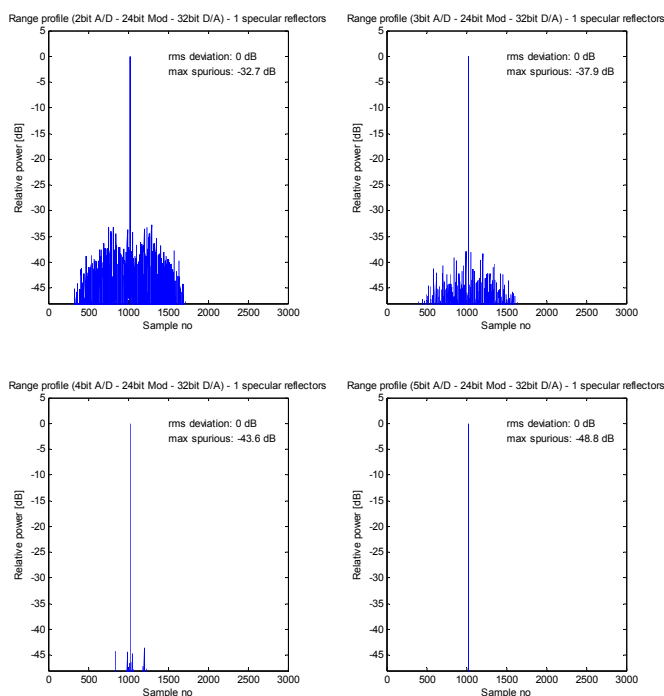
I analysen er det ikke fokusert spesielt på antall bit i datalagringen i DRFM, da deres bitbredde naturlig følger A/D-konverternes bitbredde.

7.7 Testresultater

Modelleringsresultatene fra analysen frembrakte et stort antall målekurver. Dette kapitlet skal prøve å trekke fram det vesentlige og presentere dette i noen få sammenstillinger av kurver.

7.7.1 A/D-konvertere

Første del av analysen har fokus på A/D-konverterne og deres oppførsel ved varierende bit-oppløsning. Figur 7.3 viser referansesignal 1 påtrykket datamodellen med oppløsninger mellom 2 og 5 bit. Kurvene viser klart at det maksimale spuriøsnivå blir mindre og mindre etter hvert som antall bit blir økt. Spuriøsnivået består hovedsaklig av digitaliseringsstøyen fra A/D-konverterne.



Figur 7.3 Digitaliseringsstøy ved 2, 3, 4 og 5 bit oppløsning i A/D-konverter

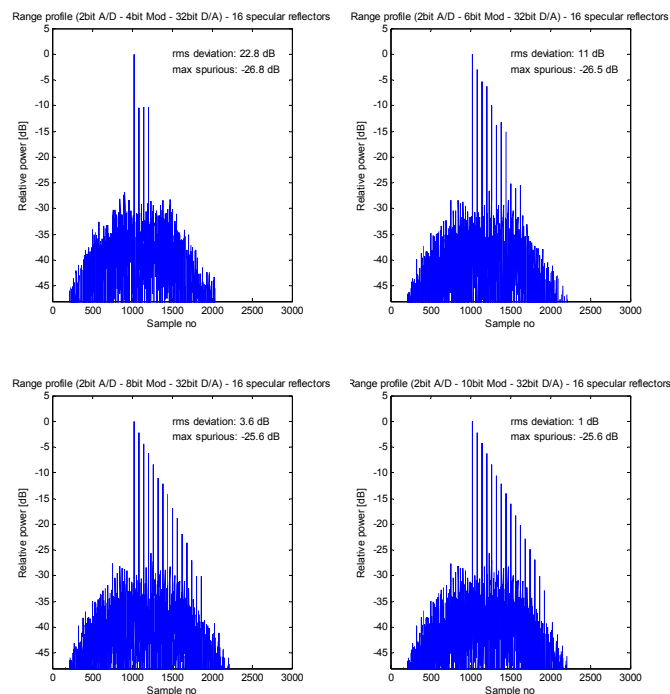
Signal-til-støyforholdet hos en A/D-konverter er gitt av 6 dB pr antall bit diskretisering som benyttes. Det er det totale støynivået som inngår i tallverdien for signal-til-støyforholdet. Derfor er det maksimale spuriøsnivået lavere enn 6 dB pr bit diskretisering. Kurvene i Figur 7.3 inneholder beregninger av rms-avviket og det maksimale spuriøsnivået, men inneholder verken det totale støynivået eller signal-til-støyforhold beregninger.

Spuriøsnivået har også et bidrag fra mangelfull sidelobeundertrykking i pulskompresjonsprosessen, men dette er først fremtredende ved høyere bit-oppløsninger. Det må bemerkes at spuriøsnivået etter pulskompresjon vil være en begrensende faktor når det gjelder refleksjoner med stor dynamikk. For at svake refleksjoner ikke skal skjermes eller forstyrres, må spuriøsnivået være tilstrekkelig lavt. Gjerne 10 dB under svakeste reflektor, men dette er avhengig av hvor presist reflektorene skal fastsettes.

7.7.2 Modulatorens amplitudemodulasjon

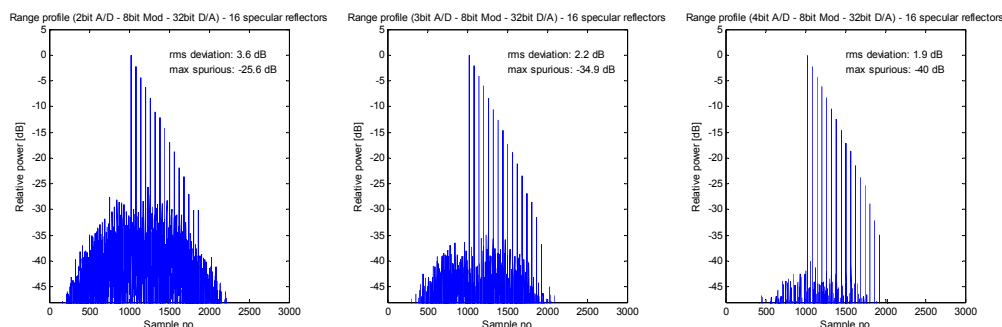
Andre del av analysen har fokus på modulatoren og derav D/A-konverternes dimensjonering. Først ses det på evnen til amplitudemodulasjon ved forskjellige bit-oppløsninger. Figur 7.4 viser referansesignal 2 påtrykket datamodellen med modulatoroppløsningene 4, 6, 8 og 10 bit. Alle kurvene i Figur 7.4 er beregnet med A/D-konverteroppløsning på 2 bit og D/A-konverteroppløsning på 32 bit, dvs at det kan ses bort fra D/A-konverteren i denne sammenheng. Modulatoren er i alle forsøkene innenfor sitt dynamiske område, dvs ingen klipping finner sted.

Figur 7.4 viser at de enkelte reflektorene kan plasseres med god presisjon så lenge antall bit i modulatorens amplituderegninger er tilstrekkelig. I praksis kan imidlertid det høye spuriøsnivået som ses i Figur 7.4 kreve større oppløsning fra A/D-konverterne enn 2 bit.



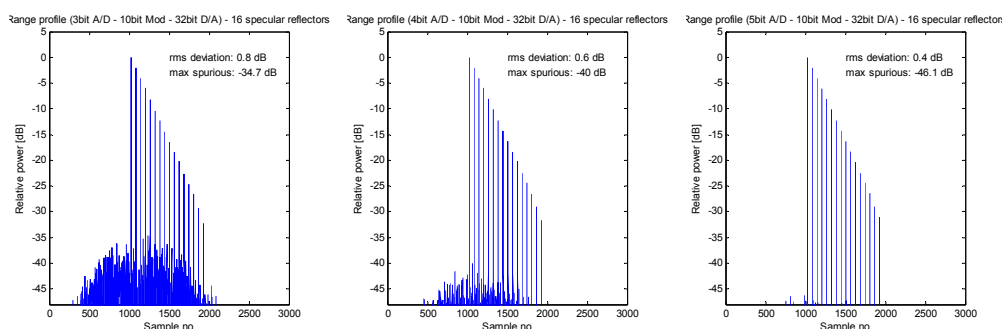
Figur 7.4 Forskjellig amplitudepresisjon i modulator ved 2 bit A/D-konvertere

Som vist i Figur 7.5 er modulatorens evne til å plassere de enkelte reflektorene presist i amplitude, kun gitt av amplituderegningenes oppløsning i modulatoren. Dette blir bekreftet av at rms-avviket holdes tilnærmet konstant ved varierende A/D-konverteroppløsning. Det varierende spuriøsnivået i de forskjellige kurvene påvirker imidlertid nøyaktigheten til de enkelte reflektorene. Dette høyner rms-avviket spesielt på kurven med 2 bit A/D-konverterne. Beregningene av rms-avviket utføres på de enkelte reflektoramplitudene i forhold til ønsket amplitude gitt av referansesignalet. På tilsvarende måte kan dette vises for 10 bit presisjon i modulatoren.



Figur 7.5 8 bit amplitudepresisjon i modulator ved forskjellig A/D-konverteroppløsning

Ved 10 bit presisjon, Figur 7.6, er også rms-avviket tilnærmet konstant ved forskjellige A/D-konvertere. Det kan også ses at 10 bit presisjon gir generelt et rms-avvik mindre enn 1 dB mot rundt 2-3 dB som kan oppnås ved 8 bit presisjon.

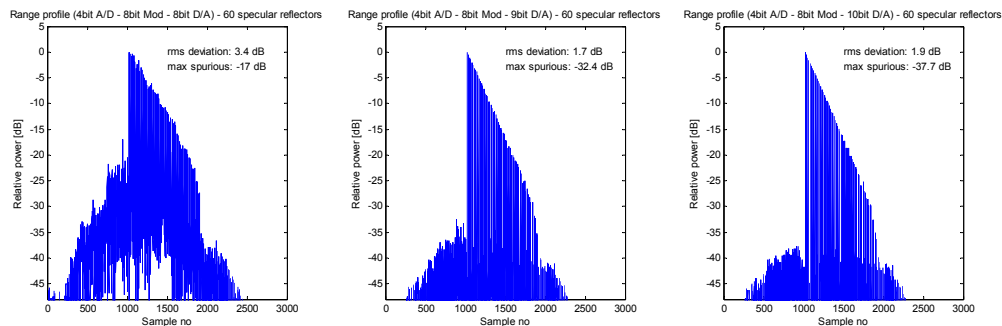


Figur 7.6 10 bit amplitudepresisjon i modulator ved forskjellig A/D-konverteroppløsning

7.7.3 Modulatorens addisjonsnettverk og D/A-konvertere

Den neste viktige oppgaven til modulatoren er dens evne til å summere et stort antall samtidige reflektorer. Referansesignal 3 benyttes for å sette addisjonskretsene på prøve mens forskjellig oppløsninger i addisjonen utprøves. Under forsøket benyttes 4 bit A/D-konvertering og 8 bit amplitudemodulasjon, men resultatene kan omregnes til andre konfigurasjoner. Antall bit i D/A-konverterne følger av antall bit i addisjonen.

I Figur 7.7 kan, fra venstre, resultatene fra 8, 9 og 10 bit addisjon ses. Kurven for 8 bit addisjonen har kraftige spuriøse komponenter og store rms-avvik. Dette skyldes at addisjonskretsene overstyres og de sinusformede signalene klippes, noe som genererer 2.harmoniske komponenter som gir opphav til de spuriøse signalene under pulskompresjonsprosessens konvolusjon. Ved 9 bit addisjon er de spuriøse komponentene dempet en god del, men addisjonskretsene blir enda overstyrt. Samtidig som det spuriøse nivået er dempet er også rms-avviket blitt lavere. Den siste kurven til høyre i Figur 7.7 viser addisjon med 10 bit presisjon. Her er både spuriøsnivået og rms-avviket lavt. Videre økning i addisjonskretsenes bit-antall vil ikke bedre gjengivelsen ytterligere, da ingen informasjon er gått tapt i addisjonsprosessen.



Figur 7.7 Resultat fra addisjon av mange samtidige reflektorer gitt antall bit i addisjonen

7.8 Diskusjon

I dette kapitlet vil de enkelte resultatene fra analysen bli diskutert. Den omtrentlig samme tematiske oppbygning vil bli brukt.

7.8.1 A/D-konvertere

Avveiningene for og mot mange bits representasjon i konverterne kan stort sett gjøres ut fra ønsket om dynamikk i radarscenen. Spuriøsnivået til den sterkeste av reflektorene vil i praksis være bestemmende for den totale dynamikken.

Få bit i konverterne gjør implementasjonen av høyhastighetskretsene enklere, men kun et begrenset dynamisk område oppnås. Det må også legges til at ved færre bit ut fra konverterne, så blir bitbreddene gjennom hele konstruksjonen smalere, med alle de besparelsene det fører med seg.

Hvis det velges en høyere bit-oppløsning i A/D-konverterne, så oppnås en bedre grad av 'troverdighet' i gjenskapelsen av de reflekterte radarsignalene i tillegg til det lave spuriøsnivået. Bare det at spuriøsnivået er lavt, gjør at de kunstige radarrefleksjonene ikke umiddelbart skiller seg fra naturlige refleksjoner.

Resultatene fra analysen viser at spuriøsnivået etter pulskompresjonen i radaren er lavere enn spuriøsnivået inn på radarantennen, pga pulskompresjonens undertrykking av andre pulskoder enn dens egen. Dette gjør at marginen mellom laveste ønskede reflektoramplitude og spuriøsnivået fra A/D-konverterne kan minskes om nødvendig for å spare implementasjonskostnader. Det må også nevnes at i tillegg til spuriøsnivået fra DRFM digitaldel så kommer analogdelens bidrag, i form av mikrobølge opp/ned-konverter med et forholdsvis begrenset dynamisk område.

7.8.2 Referansesignalene

Når det gjelder modulasjon av radarsignalene i tid og amplitude, så kan det innledningsvis diskuteres om referansesignalene som er benyttet i analysen er representative for de

refleksjonene som finnes i den virkelige verden, både av naturlige og menneskelagde objekter. Som tidligere nevnt er ikke disse refleksjonene fra noe objekt med naturlig form, men likevel kan det hevdes at refleksjonene *kunne* stamme fra et slikt objekt. Kravene til lett gjenkjennbar form har veid tyngst i valget av referansesignaler.

Den dynamikken som er lagt inn mellom de enkelte reflektorene i referansesignalene vil nok i en del tilfeller være unaturlig å benytte som mal ved en implementasjon. Men, resultatene fra analysen kan lett skaleres til å gjelde andre inngangsparametere. Det samme gjelder antall reflektorer. I noen tilfeller vil et færre antall være naturlig å bruke ved dimensjoneringen, mens i andre tilfeller vil et langt større antall være foretrukket. Fordelingen mellom sterke og svake reflektoramplituder kan også være en annen inngangsparameter. Referansesignalene som her er benyttet har uniform amplitudfordeling. Hvis en større andel av reflektorene har store amplituder eller reflektorenes dynamiske spenn minskes vil dette føre til behov for flere bit oppløsning i addisjonsnettverket.

7.8.3 Modulatorens amplitudemodulasjon

Amplitudemodulatorens presisjon er gitt av antall bit i amplituderegningene. I mange tilfeller vil ikke kravet til denne nøyaktigheten være så stor, og da kan antall bit i beregningene spares. Det er her viktig å skille antall bit i A/D-konverterne som bestemmer spuriøsnivået og antall bit i amplitudemodulatoren som bestemmer hvor nøyaktig refleksjonspunktene kan fastsettes.

7.8.4 Modulatorens addisjonsnettverk og D/A-konvertere

Presisjonen i modulatorens addisjonsnettverk og D/A-konvertere, dvs hva er differansen i antall bit mellom amplitudemodulatoren og addisjonsnettverket, er koblet til hvor mange og hvor sterke reflektorer som skal kunne håndteres samtidig. Med samtidighet menes her varigheten av en radarpuls eller sagt på en annen måte, den tiden en radarpuls belyser et objekt. På høyoppløselige radarer har imidlertid pulsene lang varighet og mange refleksjonspunkter vil derfor bli belyst samtidig. Av den grunn er deres antatte amplitudfordeling og dynamikk av interesse for dimensjoneringen av addisjonsnettverket og antall bit i D/A-konverterne. Generelt forbrukes maksimalt 1 bit ekstra for hver dobling i antall reflektorer som skal adderes, men pga at refleksjonene typisk har forskjellige amplitude og fase vil det reelle behovet være langt lavere.

7.9 Konklusjon

Denne analysen viser at en DRFM med en egnet modulator kan generere et stort antall troverdige radarrefleksjoner med stor presisjon. Med 4 bit A/D-konvertere, 10 bit D/A-konvertere, og en modulator med 8 bit amplitudemodulasjonen og 10 bit addisjonsnettverk, vil den typisk kunne generere mer enn 100 overlappende refleksjoner med amplitudenøyaktighet bedre enn 2 dB og dynamikk på 30 dB.

8. Analyse av mulige modulatorimplementasjoner for EKKO II

8.1 Innledning

Dette kapitlet tar for seg hvordan modellering av radarrefleksjoner kan implementeres i den digitale modulator for den eksperimentelle radarjammeren EKKO II. Inntil nå har modulatorene kun vært planlagt med direkte realisert modellering. Ved hjelp av resultatene fra denne avhandlingen vil også modulatorene kunne bli implementert med beregnet realisering, som i mange tilfeller har store fordeler fremfor direkte realisert modellering.

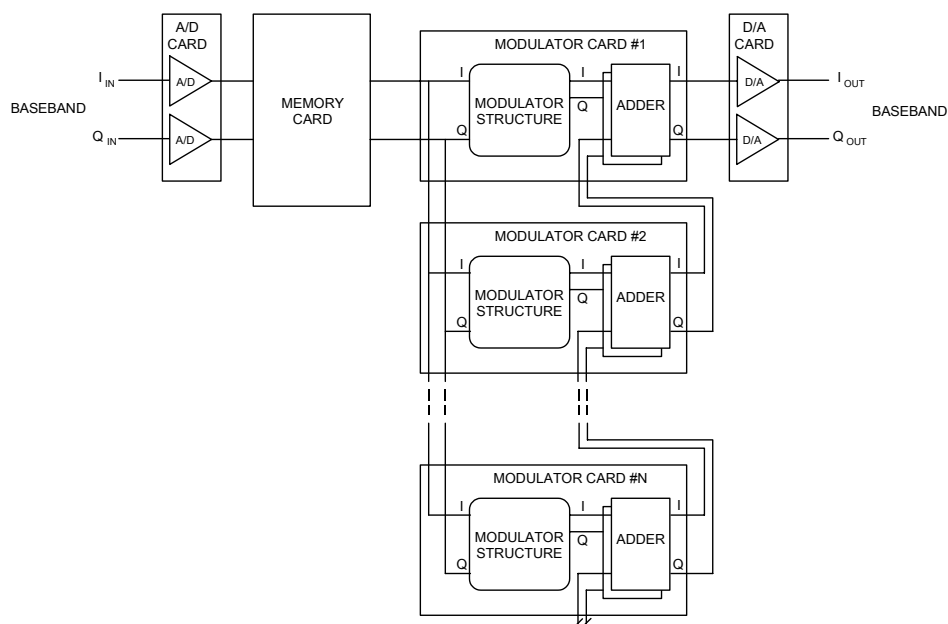
EKKO IIs digitale modulator er planlagt bygget opp med en rekke FPGA-kretser som hver for seg kun inneholder en del av strukturen for modulatorene. Avhengig av aktuell situasjon vil forskjellige konfigurasjoner lastes ned i FPGA-kretsene. Av den grunn er det viktig at modulatorens maskinvare er bygget opp slik at de forskjellige strukturene kan innpasses uten problemer. Dermed er valg av FPGA-plattform og hvordan disse kretsene kobles sammen viktig.

På grunn av at de foreslåtte realiseringskonseptene for beregnet realisering bygger på modellering i frekvensdomenet, blir tilgjengelige implementasjoner for hurtig DFT tatt opp i dette kapitlet. Analysen vil omfatte både vanlig tilgjengelige FFT / IFFT *Intellectual Property* (IP) fra FPGA-leverandøren og de mer spesialiserte IP-kjernene fra uavhengige leverandører. Det er vurdert som lite tids- og kostnadseffektivt å gjennomføre selve utviklingen av IP-kjernen lokalt, da dette er en meget krevende oppgave siden kravene til kjernens hastighet er svært stor.

Formålet med analysen i dette kapitlet blir dermed å sammenligne tilgjengelige FPGA-kretser og IP-kjerner, og foreslå en mulig FPGA sammenkoblingsstruktur som kan håndtere de forskjellige modulatorstrukturene. Analysen baserer seg på datablad og opplysninger fra leverandørene av både FPGA-kretser [28] og IP-kjerner [28], [31] - [33].

8.2 Modulatorens strukturelle oppbygning i EKKO II

Den totale modulatorstrukturen i EKKO II er planlagt implementert på fire identiske kretskort. Kun programmeringen av kortenes FPGA-kretser vil kunne avvike. Hvert av kortene er sammenkoblet i en parallell struktur, ref Figur 8.1, hvor inngangssignalene fra datahukommelsekortet fordeles direkte til alle modulatkort mens utgangssignalene fra modulatorene summeres til et resultatsignal før overføring til D/A-kortet. Denne strukturen tillater at de enkelte modulatkortene lastes med forskjellig konfigurasjon hvis ønskelig. På denne måten håndterer systemet at et modulatkort er implementert med direkte realisert modellering, et annet kort med beregnet realisert modellering osv – responsen fra de enkelte modulatkortene blir i alle tilfeller addert i tidsdomenet. Figur 8.1 viser en skisse av denne strukturen.



Figur 8.1 Sammenkobling av modulatkort i EKKO IIs DRFM-kjerne

Bakplanet i EKKO IIs digitaldel er tilrettelagt for denne sammenkoblingsformen og har i dagens versjon plass til 4 modulatkort. I prinsippet kan systemet utvides til å håndtere flere modulatkort ved behov.

8.3 Valg av FPGA for modulatkortene

Det enkelte modulatkort må inneholde den standardiserte addisjonskjeden som forbinder kortenes utganger sammen i tillegg til diverse kontroll og styringslogikk som alle kortene i systemet har. Resterende areal på kortet kan deretter tilegnes FPGA-kretser for selve modulaturstrukturen.

Modulatkortene skal bestykes med et antall moderne FPGA-kretser av fabrikat Xilinx, Inc. Valget av FPGA-fabrikat er tatt på et tidligere tidspunkt under utviklingen av EKKO II og gjelder alle kretskortene i systemet. Det er viktig å basere utviklingen på en plattform i stedet for å legge tidkrevende innsats i flere parallelle aktiviteter. Opprinnelig var også FPGA-typene, XCV300E og XCV2000E fra Xilinx Virtex-E familien bestemt for hele systemet. Disse kretsene er benyttet på A/D-kortet, datahukommelsekortet og D/A-kortet, men for de gjenstående modulatkortene er det aktuelt å velge en mer moderne FPGA-krets siden utviklingen har gått et langt steg videre siden de første kortene i EKKO II systemet ble påbegynt. For å dra mest mulig nytte av tidligere erfaringer med Virtex-E familien, vil en av dens etterfølgende familier bli valgt.

Valget av FPGA-krets er styrt av at kretskortene både skal kunne implementere direkte realisert modellering og beregnet realisert modellering, i tillegg til eventuelle nye algoritmer. Denne mangedelte bruken av den fysiske maskinvaren er et kjennetegn for all reprogrammerbar logikk, og med de store FPGA-kretsene som her er aktuelle kan hele system implementeres på denne måten.

Hvor mange FPGA-kretser som kan plasseres på et enkelt modulorkort er avhengig av en rekke faktorer; FPGA-kretsens fysiske størrelse, deres effektforbruk, hvor mange forbindelser som skal legges mellom de enkelte kretsene, skal alle kretsene ha forbindelser til alle de andre kretsene, trenger forbindelsene termineringer og har eventuelt kretsen innvendig støtte for dette, har kretsene så mye innvendig hukommelse at det ikke trengs utvendige RAM-kretser, osv.

Som tidligere nevnt i dette kapittel er det et ønske at den valgte FPGA-krets kommer fra en etterfølgende familie av Virtex-E familien. Følgende familier er dermed aktuelle; Virtex-II, Virtex-II Pro og den hittil siste på stamtavlen, Virtex-II Pro X. Tabell 8.1 viser nøkkelparametere for kretsene i disse familiene. Det er hovedsaklig de største tilgjengelige kretsene som er av interesse for modulorkortene.

		CLB Resources			Memory Resources			DSP	Clock Resources	I/O Features			Speed									
Virtex-II Series EasyPath Solutions (see note 4)		System Gates (see note 1)			Max. Distributed RAM Bits (kbits)				DCM Frequency (min/max)	# DCM Blocks (see note 3)	Digitally Controlled Impedance	Maximum Differential I/O Pairs	Commercial Speed Grades (slowest to fastest)	Industrial Speed Grades (slowest to fastest)	Configuration Memory (Bits)	RocketIO™ Transceiver Blocks (3.125 Gbps)	RocketIO™ X Transceiver Blocks (10.3125 Gbps)	PowerPC™ Processor Blocks				
		CLB Array (Row x Col)	Number of Slices	Logic Cells (see note 2)	CLB Flip-Flops	# 18 kbits Block RAM	Total Block RAM (kbits)	# 18x18 Dedicated Multipliers				Maximum I/O	I/O Standards									
Virtex-II Pro Family – 1.5 Volt																						
XC2VP2	*	16 x 22	1,408	3,168	2816	44	12	216	12	24/420	4	YES	100	204	LDT-25, LVDS-25, LVDSX-25, BLVDS-25, ULVDS-25, LVPECL-25, LVCMOS25, LVCMOS18, LVCMOS15, PCI33, LVTTL, LVCMOS33, PCI-X, PC166, GTL, GTL+, HSTL I (1.5V), 1.8V, HSTL II (1.5V), 1.8V, HSTL III (1.5V), 1.8V, HSTL IV (1.5V), 1.8V, SSTL2I, SSTL2II, SSTL18 I, SSTL18 II	-5 -6 -7	-5 -6	1.3M	4	0		
XC2VP4	*	40 x 22	3,008	6,768	6,016	94	28	504	28	24/420	4	YES	172	348		-5 -6 -7	-5 -6	3.0M	4	1		
XC2VP7	*	40 x 34	4,928	11,088	9,856	154	44	792	44	24/420	4	YES	196	396		-5 -6 -7	-5 -6	4.4M	8	1		
XC2VP20	*	56 x 46	9,280	20,880	18,560	290	88	1,584	88	24/420	8	YES	276	564		-5 -6 -7	-5 -6	8.2M	8	2		
XC2VP30	XCE2VP30	80 x 46	13,696	30,816	27,392	428	136	2,448	136	24/420	8	YES	372	644		-5 -6 -7	-5 -6	11.3M	8	2		
XC2VP40	XCE2VP40	88 x 58	19,392	43,632	38,784	606	192	3,456	192	24/420	8	YES	396	804		-5 -6 -7	-5 -6	15.5M	0 or 12	2		
XC2VP50	XCE2VP50	88 x 70	23,616	53,136	47,232	738	232	4,176	232	24/420	8	YES	420	852		-5 -6 -7	-5 -6	19.0M	0 or 16	2		
XC2VP70	XCE2VP70	104 x 82	33,088	74,448	66,176	1,034	328	5,904	328	24/420	8	YES	492	996		-5 -6 -7	-5 -6	25.6M	16 or 20	2		
XC2VP100	XCE2VP100	120 x 94	44,096	99,216	88,192	1,378	444	7,992	444	24/420	12	YES	572	1,164		-5 -6 -7	-5 -6	33.5M	0 or 20	2		
XC2VP125	XCE2VP125	136 x 106	55,616	125,136	111,232	1,738	556	10,008	556	24/420	12	YES	644	1,200		-5 -6 -7	-5 -6	42.7M	0, 20 or 24	4		
Virtex-II Pro X Family – 1.5 Volt																						
XC2VPX20	*	56 x 46	9,792	22,032	19,584	306	88	1,584	88	24/420	8	YES	276	556	Same as above	-5 -6 -7	TBD	8.05M	0	8	1	
XC2VPX70	XCE2VPX70	104 x 82	33,088	74,448	66,176	1,034	308	5,544	308	24/420	8	YES	492	996	Same as above	-5 -6 -7	TBD	25.60M	0	20	2	
Virtex-II Family – 1.5 Volt																						
XC2V40	40K	8 x 8	256	576	512	8	4	72	4	24/420	4	YES	44	88	LDT-25, LVPECL-33, LVDS-33, LVDS-25, LVDSX-33, LVDSX-25, BLVDS-25, ULVDS-25, LVTTL, LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, PCI33, PC166, GTL, GTL+, HSTL I, HSTL II, HSTL III, HSTL IV, SSTL2I, SSTL2II, SSTL18 I, SSTL18 II, AGP, AGP-2X	-4 -5 -6	-4 -5	0.4M	-	-	-	
XC2V80	80K	16 x 8	512	1,152	1,024	16	8	144	8	24/420	4	YES	60	120		-4 -5 -6	-4 -5	0.6M	-	-	-	
XC2V250	250K	24 x 16	1,536	3,456	3,072	48	24	432	24	24/420	8	YES	100	200		-4 -5 -6	-4 -5	1.7M	-	-	-	
XC2V500	500K	32 x 24	3,072	6,912	6,144	96	32	576	32	24/420	8	YES	132	264		-4 -5 -6	-4 -5	2.8M	-	-	-	
XC2V1000	1M	40 x 32	5,120	11,520	10,240	160	40	720	40	24/420	8	YES	216	432		-4 -5 -6	-4 -5	4.1M	-	-	-	
XC2V1500	1.5M	48 x 40	7,680	17,280	15,360	240	48	864	48	24/420	8	YES	264	528		-4 -5 -6	-4 -5	5.7M	-	-	-	
XC2V2000	2M	56 x 48	10,752	24,192	21,504	336	56	1,008	56	24/420	8	YES	312	624		-4 -5 -6	-4 -5	7.5M	-	-	-	
XC2V3000	XCE2V3000	3M	64 x 56	14,336	32,256	28,672	448	96	1,728	96	24/420	12	YES	360	720		-4 -5 -6	-4 -5	10.5M	-	-	-
XC2V4000	XCE2V4000	4M	80 x 72	23,040	51,840	46,080	720	120	2,160	120	24/420	12	YES	456	912		-4 -5 -6	-4 -5	15.7M	-	-	-
XC2V6000	XCE2V6000	6M	96 x 88	33,792	76,032	67,584	1,056	144	2,592	144	24/420	12	YES	552	1,104		-4 -5 -6	-4 -5	21.9M	-	-	-
XC2V8000	XCE2V8000	8M	112 x 104	46,592	104,832	93,184	1,456	168	3,024	168	24/420	12	YES	554	1,108		-4 -5	-4 -5	29.1M	-	-	-

Tabell 8.1 Nøkkelparametere for aktuelle Xilinx-familier, ill. fra [16]

I Virtex-II familien er XC2V6000 den største tilgjengelige kretsen pr mai 2004. Hvis antall slices brukes som sammenligningsgrunnlag, har denne kretsen vesentlig mindre intern hukommelse og dedikerte multiplikatorer enn de andre nyere familiene med tilsvarende størrelse. Til gjengjeld har kretsen noen flere I/O-pinner og klokkelinjer enn de andre. Familien har heller ikke støtte for høyhastighets serielle I/O-transceivere. Disse transceiverne kan være nyttige ved sammenkobling av flere kretser.

Virtex-II Pro familien er en nyere familie enn Virtex-II. Virtex-II Pro har pr mai 2004 kretsene XC2VP70 og XC2VP100 som de største tilgjengelige. Disse kretsene har blant annet to dedikerte PowerPC prosessorer internt til diverse sekvensielle oppgaver som ikke trenger maskinvare ytelse. XC2VP70 kretsen er sammenlignbar i størrelse med XC2V6000

fra Virtex-II familien, men overgår denne som nevnt på blant annet intern hukommelse og dedikerte multiplikatorer. Virtex-II Pro kretsene er også tilgjengelig i raskere hastighetsvarianter enn Virtex-II og er dessuten priset lavere enn Virtex-II for tiden. I alle tilfeller er prisene svært høye for disse store FPGA-kretsene; flere tusen US\$ pr stykk(!).

Den nyeste familien Virtex-II Pro X er foreløpig kun planlagt i to varianter. Denne familien er nærmest en utvidelse av Virtex-II Pro familien da Virtex-II Pro X inneholder et antall 10 Gbps serielle I/O-transceivere i stedet for 3 Gbps transceivere. Den største kretsen, XC2VPX70 har 20 slike fulldupleks transceivere på bekostning av noen få blokker med intern hukommelse og noen få dedikerte multiplikatorer. 20 transceivere av 10 Gbps utgjør en stor I/O-kapasitet i tillegg til de vanlige I/O-pinnene, og kan benyttes til FPGA til FPGA kommunikasjon innen et kretskort eller til kretskort til kretskort kommunikasjon. Prismessig er XC2VPX70 oppgitt til å bli sammenlignbar med XC2VP70.

Konklusjonen på denne korte vurderingen av valg av ny FPGA-plattform er at Virtex-II Pro Xs XC2VPX70 best vil fylle kravene til en FPGA som skal benyttes til massiv digital signalbehandling og som har store krav til I/O-kapasitet. Ved de innebygde prosessorene vil også kretsen ha betydelig kapasitet mot sekvensielle problemstillinger. 10 Gbps serielle I/O-transceivere kan benyttes både internt på kretskortene og eksternt, og har i motsetning til Virtex-II Pro seriens 3 Gbps transceivere nok kapasitet til å overføre de store mengdene data som flyter gjennom en moderne radarjammer. XC2VPX70 har i tillegg betydelig regnekapasitet med sine 308 maskinvare multiplikatorer og har stor lagerkapasitet med 5,5 Mbit RAM-hukommelse. Kretsen har 996 I/O-pinner for generelt bruk med programmerbart valg av I/O-standard. Flere av I/O-standardene innbefatter termineringer internt i FPGA-kretsen. Prismessig er også XC2VPX70 gunstig når først en FPGA med dens ytelse vurderes.

8.4 Modulatorkortets oppbygning

Modulatorkortene skal i størst mulig grad bestå av moderne kraftige FPGA-kretser som er velegnet for digital signalbehandling. Antallet kretser skal ikke velges ut i fra en viss applikasjon, men ut i fra hvor mange kretser som kan plasseres på kortene. Dette gjøres for å kunne legge inn nye større algoritmer i fremtiden. Kostnadene pr kort vil naturlig øke jo flere FPGA-kretser som legges inn. Men siden kun et lite antall kort skal produseres vil totalkostnadene i alle tilfeller være høye og kostnadene blir dermed mindre avhengig av FPGA-kretsenes pris og antall enn om kortene skulle masseproduseres.

Hvis det tas utgangspunkt i den foreslåtte FPGA, XC2VPX70, så er dens fysiske størrelse 42,5 x 42,5 mm. Tilgjengelig areal på modulatorkortene tilsier at 6 slike FPGA-kretser kan plasseres i tillegg til diverse RAM-hukommelser, som kretskortet bør være bestykket med for å være forberedt til forskjellige aktuelle strukturer. Hukommelseskretsene må kunne lagre alle samplene i et radarsignal med maksimal lengde og alle impulsresponskoeffisientene for en maksimal avstandsprofil som et minimum. Som nevnt i kapittel 8.2 må det enkelte modulatorkort også inneholde den standardiserte addisjonskjeden som knytter kortenes utganger sammen før overføring til D/A-kortet, samt diverse kontroll og styringslogikk for kortet.

Et modulorkort med en slik kompleksitet av FPGA-kretser, hukommelseskretser og annen logikk vil ha et betydelig effektforbruk. Spesielt vil FPGA-kretsene som prosesserer data med høy klokke rate trenge påmonterte kjølefinner i tillegg til luftkjøling fra kraftige vifter. Leverandørene av FPGA-kretser kan levere programvare for estimering av et systems effektforbruk ved implementering av en gitt FPGA-kode i deres kretser, ref [28]. Ulempen er at FPGA-koden vanligvis på langt nær er ferdig når kretskortet og kjølesystemet skal konstrueres, og dermed blir effektestimaterne lite nøyaktige.

Strukturen av FPGA-kretser på modulorkortet har et betydelig behov for I/O-kapasitet. Bakplanet i EKKO II digitaldel har støtte for dette med totalt 232 datalinjer inn og ut fra hvert modulorkort. Dette antallet fremkommer ved å summere antall bit inn og antall bit ut, og multiplisere med de 4 tidsmultipleksede kanalene som EKKO II systemet har.

Tabell 8.2 viser regnestykket for I/O antallet.

Beskrivelse:	Beskrivelse antall bit:	Sum antall bit:	Sum antall bit etter tidsmultipleks:
Antall bit <i>inn</i> for radarsignal	4 bit I + 4 bit Q + 1 bit kontroll	9 bit	36 bit
Antall bit <i>ut</i> for radarsignal (til neste modulorkort)	4 bit I + 4 bit Q + 1 bit kontroll	9 bit	36 bit
Antall bit <i>ut</i> fra addisjonskjeden	10 bit I + 10 bit Q	20 bit	80 bit
Antall bit <i>inn</i> fra neste modulorkorts addisjonskjede	10 bit I + 10 bit Q	20 bit	80 bit
Totalt		58 bit	232 bit

Tabell 8.2 EKKO II modulorkortenes I/O-behov

Sammenkoblingen av FPGA-kretsene på modulorkortet kan utføres på flere måter; alle signaler til alle kretser, kretsene kobles i trestruktur, noen kretser dedikeres til I/O mens andre kretser dedikeres til andre oppgaver, osv. De forskjellige alternativene har alle sine fordeler, men hva vil være mest generelt for fremtidige modulatorrealiseringer? Og hvor mye av fremtidig I/O-behov kan baseres på de serielle høyhastighets transceiverne?

Så lenge modulorkortene brukes med dagens bakplan i EKKO II, vil I/O-behovet – og mulighetene – være begrenset av bakplanets kapasitet. Selv om bakplanet oppdateres en gang i fremtiden vil ikke I/O-kapasiteten kunne økes, da maksimalt antall signaler er benyttet innenfor spesifikasjonen av kontaktene i cPCI-standarden. EKKO II digitaldel baserer seg på cPCI-standarden når det gjelder kortstørrelse, plassering av kortkontaktene og elektrisk grensesnitt mot PCI-kontroller over en 32 bit PCI-buss.

Høyhastighetskommunikasjonen mellom de enkelte EKKO II kretskortene foretas ved hjelp

av punkt til punkt forbindelser via dedikerte kortkontakter som er reservert i cPCI-standarden til diverse utvidelser og egendefinerte busser.

Hvert av modulatkortene har i dag et I/O-behov på 232 bit ganger 125 MHz rate som er hastigheten i EKKO IIs punkt til punkt forbindelser. Totalt I/O-behov blir dermed 29 Gbps for et enkelt modulatkort og 14,5 Gbps datarate mellom kortene. 14,5 Gbps kan overføres med kun 2 serielle høyhastighets transceivere som f.eks kan være koaksialkabler eller fibere. Med andre ord, et relativt lite antall transceivere kan håndtere dagens I/O-behov og ved en fremtidig utvidelse kan ytterligere transceivere tas i bruk. Hver av de foreslåtte Virtex-II Pro X FPGA-kretsene har 20 fulldupleks transceivere med en total I/O-kapasitet på hele 412,5 Gbps.

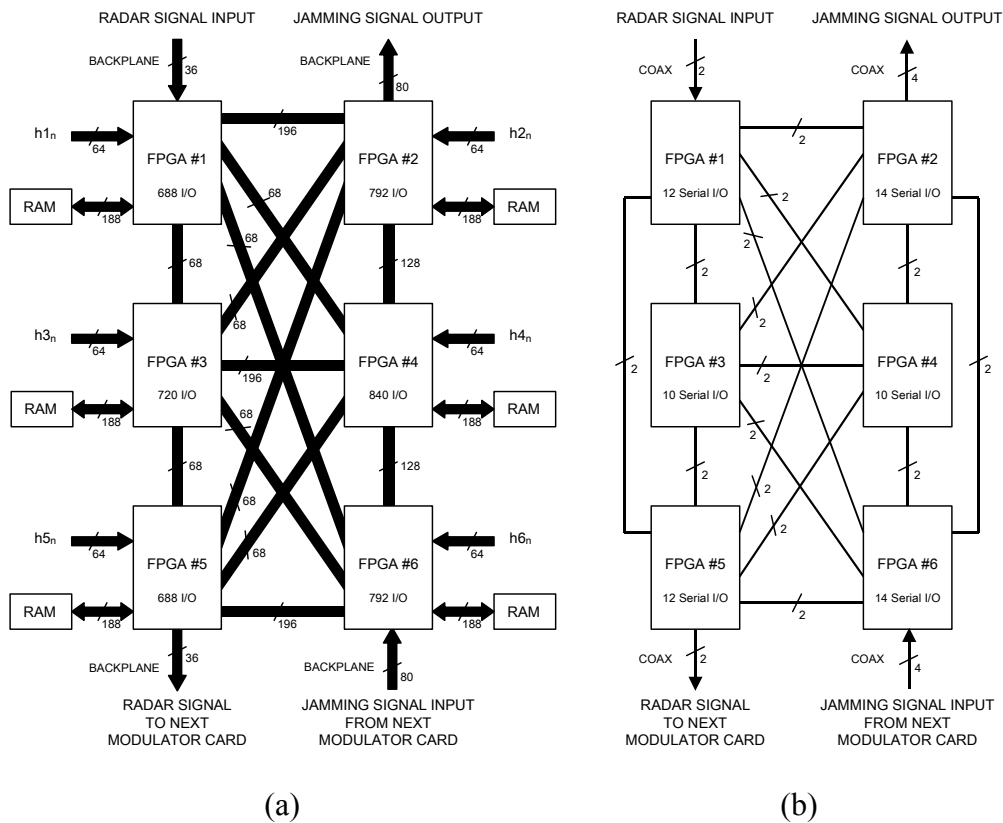
Utfordringen med høyhastighets transceivere og sanntidsdata er at informasjonskanalen tilføres en viss innskuddsforsinkelse. Vanligvis overføres informasjonen ved hjelp av en protokoll som blant annet sørger for retting av bitfeil og regenererer synkroniseringsklokken. Innskuddsforsinkelsen oppstår ved håndtering av protokollens formater og ved de innsatte FIFO-bufferne, som er satt inn for å sikre jevn overføringshastighet i kanalen. Ved å konfigurere transceiverne uten en standard protokoll kan antakeligvis mye av det ekstra tidsforbruket spares, men funksjoner som retting av eventuell bitfeil i transmisjonen faller også ut. I en del tilfeller med sanntidsprosessering vil dette være en akseptabel løsning, mens i andre tilfeller ikke. For modulatorene i EKKO II vil en viss bitfeilrate kunne aksepteres, da naturlige radarsignaler vanligvis er støybefengt og til dels ødelagt av støy. Om deler av enkelte genererte radarrefleksjoner blir ødelagt er dermed til å leve med. I sanntidsprosessering vil problemet med en viss forsinkelse i kanalen være vanskeligere å håndtere. Dette gjør at transceiverne i mange tilfeller ikke kan benyttes i tidskritiske sanntidsoppgaver.

Siden strukturen med 6 FPGA-kretser på hvert modulatkortet både skal være tilpasset dagens EKKO II konfigurasjon med direkte og beregnet modellering, en eventuell fremtidig oppdatering av systemets ytelse og eventuelle nye arkitekturer, trer dermed følgende strukturforslag frem:

Modulatkortene: Internt på de enkelte modulatkortene benyttes tradisjonelle parallelle I/O-linjer for all kommunikasjon mellom FPGA-kretsene, ref Tabell 8.3, både for dagens og antatte fremtidige kommunikasjonsbehov. I tillegg benyttes serielle transceivere, ref Tabell 8.4, koblet sammen i kortutlegget for å ha ekstra kapasitet i tilfelle kommunikasjonskrevende nye arkitekturer skal implementeres. De serielle transceiverne vil ha svært høy overføringskapasitet, men vil best kunne anvendes der hvor sanntidskravene ikke er for store.

Bakplan: Dagens EKKO II bakplan med 14,5 Gbps kapasitet mellom modulatkortene benyttes som planlagt. En fremtidig utvidelse av I/O-behovet baseres på bruk av serielle transceiverne som er eksternt koblet mellom modulatkortene. Det kan aksepteres at transceiverne innfører en viss innskuddsforsinkelse her, så lenge forsinkelsen er deterministisk.

Den foreslåtte modulatorstrukturen vil langt på vei oppfylle ønskene, da logiske strukturer og oppdelinger kan implementeres innenfor den foreslåtte struktur. Figur 8.2 viser skisser over hvordan dette er tenkt utført:



Figur 8.2 Foreslått FPGA-struktur for modulorkortene, hvor
 (a) viser antall parallele I/O-linjer i de enkelte bussene.
 Bussene har overkapasitet i forhold til dagens behov.
 (b) viser antall serielle transceivere for ytterligere utvidet
 kommunikasjonskapasitet.

Figur 8.2 viser en konstruksjon med 6 FPGA-kretser, men strukturen kan utvides med samme oppbygning til å gjelde flere kretser hvis ønskelig. Signalveiene i Figur 8.2 (a) er forbindelser som foreslås lagt ut i modulorkortet mot standard I/O-pinner på FPGA-kretsene. Disse linjene vil være tilstrekkelig for direkte realisert og beregnet realisert modellering med både dagens antall bit og samplerate i A/D og D/A-konverterne, og antatte behov i fremtiden for EKKO II. Tabell 8.3 og Tabell 8.4 viser hvilke datarater strukturen har kapasitet til.

Siden EKKO II benytter tidsmultipleksing i 4 kanaler blir alle bussbreddene firedoblet i forhold til et system uten multipleksing. Tidsmultipleksingen er påkrevd for å holde busshastighetene nede på implementerbare hastigheter. EKKO II systemet benytter i dag 125 MHz datarate både internt på kretskortene og mellom kretskortene. Ved fremtidige utvidelser av antall bit og/eller samplerate foreslås det å øke dataraten internt på modulorkortene til 250 MHz for bedre utnyttelse av databussene.

Hver av FPGA-kretsene i Figur 8.2 (a) har en 64-bit buss for impulsresponskoeffisienter til konvolusjonene. Bussenes store båndbredde er valgt for å kunne oppdatere koeffisientene hurtig. Koeffisientene blir levert fra EKKO IIs teknikkgenerator, som er en del av kontrollfunksjonen i alle kretskortene i systemet. På modulorkortene vil teknikkgeneratoren være implementert i en egen FPGA-krets utenom selve modulorfunksjonen. Figur 8.2 (a) viser også grensesnittet mot 2-port RAM-

hukommelsene. Grensesnittet består av 2 x 72 bit databuss og 2 x 22 bit adresse og kontroll, totalt 188 bit.

Figur 8.2 (b) viser foreslått sammenkobling av de serielle transceiverne for fremtidige behov. Ekstern sammenkobling mellom modulatorkortene foreslås utført med koaksialkabler, mens internt på de enkelte kortene gjøres sammenkoblingen i kretskortutlegget. De serielle transceiverne vil bli brukt til mindre tidskritiske forbindelser ved en eventuell utvidelse av antall bit og/eller samplerate, ref diskusjon om serielle transceivere og sanntidskrav tidligere i dette kapittel.

FPGA-kretsene i Figur 8.2 har også påtegnet antall I/O-signaler og serielle transceivere som vil bli ført til hver enkelt FPGA ved den foreslåtte struktur. Det er fortsatt et stort antall ubenyttede I/O-pinner på FPGA-kretsene som kan utnyttes til ytterligere kapasitetsøkning for fremtidige utvidelser. Dette kan imidlertid gå på bekostning av fremkommeligheten i kretskortutlegget og er frafalt av den grunn.

Siden FPGA-kretsene er komplekse og dyre komponenter, har mange I/O-forbindelser og krever et avansert kretskortutlegg, vil det være ønskelig å montere dem i sokkel for lettere søk og feilfinning. Dette vil være et stort pluss under utviklingsfasen. Ved bruk av sokkelmontering kan også eventuelle FPGA-kretser med feil byttes ut.

Tabell 8.3 og Tabell 8.4 viser nærmere overføringskapasitetene i strukturforslaget:

Beskrivelse	Antall linjer	Kapasitet ved 125 MHz datarate	Kapasitet ved 250 MHz datarate
Kapasitet mellom FPGA 'nabopar' (f.eks FPGA #1 og #2)	196	24,5 Gbps	49,0 Gbps
Kapasitet mellom FPGA 'firegruppe' (f.eks FPGA #1, #2, #3 og #4)	68	8,5 Gbps	17,0 Gbps
Kapasitet mellom FPGA diagonalt i strukturen (f.eks FPGA #1 og #6)	68	8,5 Gbps	17,0 Gbps
Kapasitet mellom FPGA beregnet for addisjonskjeden (FPGA #2, #4 og #6)	128	16,0 Gbps	32,0 Gbps
Kapasitet for innlasting av impulsresponskoeffisienter til FPGA	64	8,0 Gbps	16,0 Gbps
Kapasitet mellom FPGA og RAM (databuss, separat buss for les/skriv)	2 x 72	9,0 Gbps	18,0 Gbps

Tabell 8.3 Kapasiteter via tradisjonelle I/O-porter i foreslått struktur

Beskrivelse	Antall linjer	Kapasitet
Kapasitet for radarsignal inn til hvert enkelt modulatkortet	2	20,6 Gbps
Kapasitet for radarsignal ut fra hvert enkelt modulatkort	2	20,6 Gbps
Kapasitet for jammesignal inn til hvert enkelt modulatkort	4	41,3 Gbps
Kapasitet for jammesignal ut fra hvert enkelt modulatkort	4	41,3 Gbps
Kapasitet mellom de enkelte FPGA-kretsene på hvert enkelt modulatkort	2	20,6 Gbps

Tabell 8.4 Kapasiteter via serielle transceivere i foreslått struktur

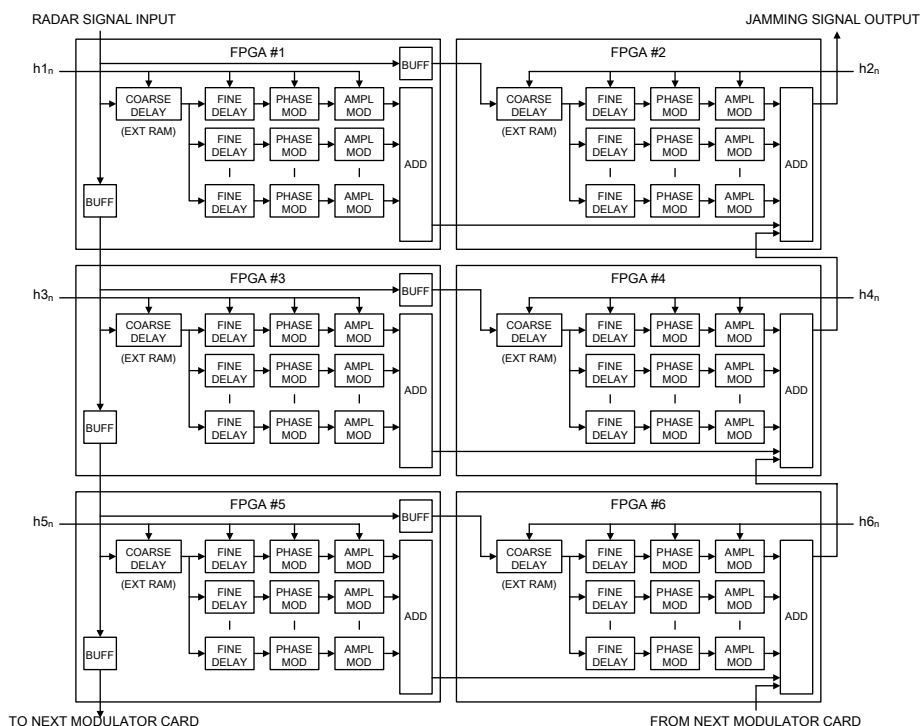
Kapasitetene som er gjengitt i Tabell 8.3 er basert på følgende antakelser om fremtidige samplerater og antall bit i A/D og D/A-konverterne: Maksimalt 1 GHz samplerate, 8 bit I / 8 bit Q for A/D-konverterne og 16 bit I / 16 bit Q for D/A-konverterne. Dette gjelder internt på de enkelte modulatkortene, mens Tabell 8.4 linje 1 – 4 som angir kapasitetene inn og ut fra modulatkortene, er beregnet for å være tilpasset kapasitetene i Tabell 8.3. Den ytterligere utvidede ytelse som er lagt inn i det foreslåtte forslag, ref Tabell 8.4 linje 5, angir en kapasitet på 20,6 Gbps ekstra mellom hver enkelt FPGA i tilfelle implementasjon av nye svært kommunikasjonskrevende algoritmer.

8.4.1 Direkte realisert modellering med foreslått FPGA-struktur

Figur 8.3 viser en skisse for hvordan direkte realisert modellering, ref kapittel 6.4, kan implementeres i modulatkort med foreslått FPGA-struktur.

Hver av FPGA-kretsene i Figur 8.3 har tilnærmet lik konfigurasjon foruten den standard fordelingen av radarsignalkopien på venstre side av strukturen og addisjonsnettverket som summerer jammesignalene langs den høyre side av strukturen. Hver av FPGA-kretsene i strukturen vil kunne lage et antall radarrefleksjoner gitt av FPGA-kretsens ressurser som hukommelse, addisjonskretser og multiplikatorer. Hver FPGA-krets er tilkoblet en ekstern RAM-hukommelse som i denne realiseringen benyttes til å plassere en gruppe av radarrefleksjoner til en gitt avstand fra radaren. Dermed trenger FPGA-kretsens interne hukommelse kun ivareta innbyrdes forsinkelse mellom reflektorene i gruppen. Den interne hukommelsen i aktuell FPGA er gruppert i 18 kbit grupper, og siden EKKO II benytter tidsmultipleksing i 4 kanaler, vil hver forsinkelsesblokk i hver modulorgren benytte 4 hukommelsesblokker. Denne forsinkelsesstørrelsen gjør at de enkelte refleksjonene maksimalt kan separeres med $(4 \times 18 \text{ kbit} / 9 \text{ bit}) \times 2 \text{ ns} = 16,384 \mu\text{s}$, hvor 9 bit er antall bit i

hvert datasample og 2 ns er sampleraten. 16,384 μ s forsinkelse tilsvarer 2,46 km, noe som er langt mer enn nødvendig.



Figur 8.3 Direkte realisert modellering i modulorkort med foreslått FPGA-struktur

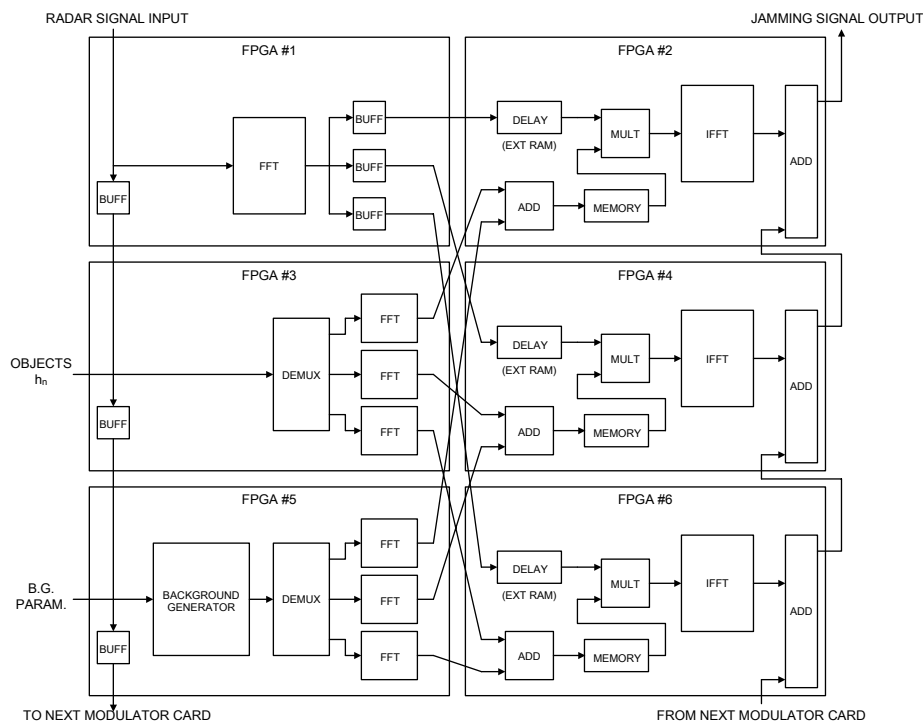
Fasemodulatoren i en modulorgren kan bygges opp av 4 addisjonskretser, en for hver tidsmultiplexet kanal. Addisjonene implementeres med vanlig FPGA-logikk som er tilgjengelig i stort antall. Amplitudemodulatoren må derimot implementeres med en konvertering fra fase til amplitudedigitalisering, og med separate multiplikatorer for I og Q-data. Totalt vil 4 hukommelsesblokker og 8 multiplikatorblokker forbrukes pr amplitudemodulator. Tilslutt skal refleksjonene fra hver modulorgren summeres sammen. Til dette benyttes som tidligere vanlig FPGA-logikk og ingen dedikerte blokker.

Til sammen kan det beregnes et ressursforbruk pr modulorgren på 8 hukommelsesblokker, 8 multiplikasjonsblokker og en del generell logikk. Teoretisk vil det dermed kunne plasseres 38,5 modulorgrener pr FPGA, ref ressurser for aktuell FPGA i Tabell 8.1, men med f.eks 85% utnyttelse kan 32 parallelle grener implementeres i praksis. Med 6 FPGA-kretser pr modulorkort kan dermed en kompleks radarrefleksjon med opptil 192 reflektorer kunne genereres med ett modulorkort og med 4 modulorkort hele 768 reflektorer. Med dette antall reflektorer kan samtlige avstandsceller i en radar med pulskompresjon opp til 768 fylles.

8.4.2 Beregnet realisert modellering med foreslått FPGA-struktur

Figur 8.4 viser en skisse for hvordan beregnet realisert modellering kan implementeres i modulorkort med foreslått FPGA-struktur. Eksemplet viser en oppkobling med distribuert FCP-konvolusjon med parametrisk generator for bakgrunnsprofil som er omtalt i kapittel

6.11. Den distribuerte konvolusjonen benytter 3 parallelle grener.



Figur 8.4 Beregnet realisert modellering i foreslått FPGA-struktur

FPGA #1, ref Figur 8.4, inneholder en FFT-transformasjon av radarsignalet. Siden modulatorens innskuddsforsinkelse er direkte avhengig av denne transformasjonens hastighet er hele FPGA-kretsen avsatt til dette, da transformasjonens hastighet og antall datapunkter bestemmer ressursbruken i FPGA-kretsen. Det er beskrevet mer om implementasjon av FFT-transformasjoner i kapittel 8.5.

FPGA #2, #4 og #6 inneholder hovedsaklig komplekse multiplikatorer og inverse FFT-transformasjoner. De inverse FFT-transformasjonenes hastighet er her like viktige for modulatorens innskuddsforsinkelse som FFT-transformasjonen i FPGA #1. De komplekse multiplikasjonene implementeres ved hjelp av de dedikerte multiplikasjonsblokkene i FPGA-kretsen, slik at disse ikke er noe hastighetsproblem. FPGA #2, #4 og #6 inneholder i tillegg addisjonskretser og hukommelseskretser. Addisjonene benyttes for å summere de enkelte frekvenssamplene fra modulasjonene av objektene med frekvenssamplene fra modulasjonene av bakgrunnsprofilen. Denne operasjonen utføres ved hjelp av standard logikk i FPGA-kretsen. Hukommelseskretsene bygges opp som 2-port hukommelser ved hjelp av FPGA-kretsens innebygde hukommelsesblokker. Hukommelseskretsene har som funksjon å lagre modulasjonssamplene slik at de ikke trengs oppdateres utenom ved forandring i modulasjonsdata. FPGA #2, #4 og #6 inneholder til slutt den vanlige addisjonskjeden for summering av jammesignalet fra de enkelte konvolusjonsgrenene.

FPGA #3 og #5 inneholder preparering av modulasjonssignalene av henholdsvis objekter og bakgrunn. I begge kretser blir modulasjonssignalet splittet i tre; først til første gren, så til neste gren og tilslutt til siste gren. I hver av grenene blir tidssamplene transformert til frekvensdomenet ved hjelp av en FFT-transformasjon. FFT-transformasjonene opptar her mindre plass i FPGA-kretsen enn i FFT-transformasjonen i FPGA #1 grunnet lavere

oppdateringsrate av modulasjonsdata enn av radarsignalet. Når transformasjonene skal yte noe mindre hastighet, vil ressursbruken minskes relativt mye mer, slik at 3 transformasjoner får plass i en enkelt FPGA. FPGA #5 inneholder i tillegg en sekvensiell generator for bakgrunnsprofilen.

Prinsippskissen i Figur 8.4 har utelatt problematikken rundt EKKO IIs 4 tidsmultipleksede kanaler. Signalene inn og ut av modulatorene må multiplekseres før signalbehandlingen med beregnet realisert modellering utføres og deretter demultiplekseres etter endt modellering.

Med den foreslåtte strukturen vil modellering av svært mange reflektorer kunne realiseres. I prinsippet kan reflektorer plasseres i hele modulatorens avstandsområde med en avstand gitt av sampleintervallet. Med avstandsområde på f.eks 100 km og samplerate lik 2 ns, vil ett enkelt modulatkort kunne modellere over 330000 refleksjonspunkter med oppløsning i avstand på 0.3 m. Om radarens utsendte bølgeform har lang varighet vil svært mange av refleksjonspunktene være overlappende i tid. En bølgeform med 50 μ s varighet vil i dette eksemplet generere 25000 overlappende reflektorer.

Et slikt antall vil imidlertid overstyre addisjonsnettverket og D/A-konverterne fullstendig om ikke utsignalene skaleres tilstrekkelig. 25000 overlappende signaler trenger en D/A-konverter med opptil 15 bit mer dynamikk enn et enkelt signal, ref ligning 7.2, avhengig av hvilke antakelser som legges til grunn for signalenes amplitudfordelinger. Skalering kan foretas siden jammeren ikke trenger å produsere et utgangssignal med større dynamikk enn det radaren normalt tar i mot.

Tilsvarende oppkoblinger som vist i Figur 8.4 kan vises for andre modulatorstrukturer. Påstanden er at de andre strukturene omhandlet i denne avhandlingen og andre fremtidige strukturer vil kunne implementeres i den foreslåtte FPGA-struktur.

8.5 IP-kjerner for FFT / IFFT prosessering

IP-kjerner for FPGA-kretser finnes for mange forskjellige applikasjoner. IP står for *Intellectual Property* og er ferdig konstruerte blokker for innbygning i konstruksjoner under utvikling. Fordelen med IP-kjerner er at disse blokkene er tilpasset og optimalisert for den aktuelle FPGA-krets av en spesialiststab av dyktige konstruktører. Selv om IP-kjernene kan være dyre i innkjøp, kan deres investering ofte forsvares med sparte arbeidstimer i prosjektet eller høyere oppnådd ytelse enn ellers.

IP-kjerner for signalprosessering finnes i stort utvalg, men antallet kjerner for høyhastighets FFT-transformasjoner er begrenset. For modulatkortene er FFT-transformasjoner kun aktuelle for beregnet realisert modellering, da direkte realisering ikke foretar transformasjon til frekvensdomenet.

De følgende avsnittene gjengir de viktigste kildene til IP-kjerner som ble identifisert. Som bakgrunninformasjon om pipeline FFT-transformasjoner er artikkelen [5] benyttet.

8.5.1 Xilinx Inc – CoreGen

FPGA-fabrikanten Xilinx Inc, ref [28], har et eget bibliotek, *CoreGen* for sine IP-kjerner som følger med utviklingsprogramvaren for deres FPGA-produkter. Biblioteket inneholder flere FFT-transformasjoner hvor de seneste versjonene er parametriske slik at ytelse, antall datapunkter, bussbredder osv kan bestemmes ved innsetting i aktuell konstruksjon. Disse kjernene er anvendelige i de fleste tilfeller, men mangler den høye ytelsen som enkelte av transformasjonene i modulorkonstruksjonen trenger.

Når høyeste ytelsesnivå velges vil et datasample prosesseres pr klokkesyklus. Maksimal klokkefrekvens gir dermed den totale prosesseringstiden. I tillegg trengs en klokkeperiode pr datasample for innlesing og for utlesing av data. Innlesing, utlesing og selve transformasjonen kan foregå som tre uavhengige prosesser slik at en kontinuerlig datastrøm kan flyte gjennom FFT-transformasjonen. Tabell 8.5 viser noen eksempler på ytelser med CoreGen FFT-transformasjoner.

Point Size	Input Data Width	Phase Factor Width	Slices	Block RAM	MULT 18x18	Max. Clock Frequency (MHz)		Transform Time			Device
						-7	-6	Clock Cycles	Time (μ s)		
									-7	-6	
64	16	16	2401	19	9	192	169	64	0.33	0.38	xc2vp20
1024	16	16	2764	19	9	197	172	1024	5.20	5.95	xc2vp20
2048	16	16	3723	22	18	194	170	2048	10.56	12.05	xc2vp20
8192	16	16	3937	76	18	176	156	8192	46.55	52.51	xc2vp30

Tabell 8.5 CoreGen FFT-transformasjon i høyttelse modus, ill. fra [28]

Tabell 8.5 oppgir som sagt kun selve transformasjonstiden. For bruk i modulatorene må imidlertid også inn og utlesingstidene tas med i regnestykket.

8.5.2 Pentek Inc – GateFlow

Pentek Inc, ref [31], er en uavhengig IP-kjerneleverandør for Xilinx FPGA-kretser. Deres hovedområde er systemkomponenter og hele system, men de leverer også noen IP-kjerner for FFT-transformasjoner.

Følgende IP-kjerner ble identifisert; enkel 4096-punkters FFT, firedobbel 1024-punkters FFT og firedobbel 4096-punkters FFT. Alle er pipeline transformasjoner.

Tabell 8.6 viser data for Pentek 4096-punkters FFT-transform modell 403. Denne IP-kjernen prosesserer ett datapunkt pr klokkesyklus slik som Xilinx CoreGen FFT-transformasjonen. Imidlertid er maksimal klokkefrekvens lavere enn for CoreGen kjernen, ref Tabell 8.5, men siden dette er en pipeline transformasjon vil utgangsdata finnes på utgangsporten straks pipelinekjeden er fylt opp. Forsinkelsen i forbindelse med pipelinekjeden vil typisk være mindre enn forsinkelsen med innlesing av alle samplene som hos CoreGen IP-kjernen. I følge databladet skal pipelinekjeden være 1104 steg lang. Ved klokkefrekvens lik 140 MHz utgjør dette en forsinkelse på 7,88 μ s. Dette er raskere enn CoreGen kjernen som både må

lese inn data og prosessere data før utgangsdata oppstår på utgangsporten.

Xilinx Speed Grade	Max Clock Rate	Max Input Rate	Core 403 FFT Time
6	140 MHz	140 MSPS	29.26 us
5	126 MHz	126 MSPS	32.51 us
4	110 MHz	110 MSPS	37.24 us

Resources	Core 403
Slice LUTs	8,955
Slice Flip Flops	7,278
Block RAM	44
Block Multipliers	28
Global Clocks	1

Tabell 8.6 Pentek single 4096-punkters FFT, ill. fra [31]

Tabell 8.7 viser data for Pentek firedoble FFT-transformasjoner med 1024 datapunkters lengde (modell 401) og 4096 datapunkters lengde (modell 404). Prosesseringstiden i følge tabellen er svært oppløftende, men ved nærmere granskning av databladet er denne prosesseringstiden kun tiden mellom hver ferdigprosessert FFT når de 4 transformasjonene er overlappet og forskjøvet. Prosesseringstiden for en komplett FFT-transformasjon er dermed 4 ganger hva som oppgis i tabellen. En 1024-punkters FFT tar 7,31 μ s og en 4096-punkters FFT tar 29,2 μ s. Altså samme ytelse som den enkle kjernen fra Pentek, ref Tabell 8.6.

Xilinx Speed Grade	Max Clock Rate	Max Input Rate	401 FFT Time	404 FFT Time
-6	140 MHz	560 MSPS	1.83 us	7.31 us
-5	126 MHz	504 MSPS	2.03 us	8.10 us
-4	110 MHz	440 MSPS	2.33 us	9.31 us

Resources	Core 401	Core 404
Slice LUTs	12,702	13,184
Slice Flip-Flops	11,410	14,590
Block RAM	40	90
Block Multipliers	64	76
Global Clocks	1	1

Tabell 8.7 Pentek firedobbel 1024 og 4096-punkters FFT, ill. fra [31]

8.5.3 Roke Manor Research

Roke Manor Research, ref [32], er et forsknings og utviklingsfirma innen avansert digital signalbehandling. De fremmer en høyhastighets pipeline FFT-kjerne utviklet i 2002 med opptil 16384-datapunkter som den gang ble reklamert med å være den største og raskeste FFT som kunne implementeres i en enkelt Xilinx Virtex-II XC2V6000 FPGA.

I følge Xilinx datablad har den foreslåtte FPGA for modulatorkortene 114 % flere multiplikatorer og hukommelsesblokker enn XC2V6000, ref Tabell 8.1. Dette skulle tilsi at en langt større FFT enn 16384-datapunkter bør kunne implementeres i den foreslåtte FPGA. Hastighetsmessig bør også den foreslåtte FPGA komme gunstig ut, da aktuell hastighetsgradering for denne kretsen er høyere enn for den varianten av XC2V6000 som Roke Manor Research har antatt.

Tabell 8.8 viser eksempler på antatte ytelser ved bruk av Roke Manor Research's FFT-kjerne:

Transform size (points)	Word widths (bits)			Slices (%)	Hardware multipliers (%)	Block RAMs (%)	Transform time (μ S)		Input sample rate (MSps)	
	Input	Internal	Twiddle				Auto P&R	Floor planning	Auto P&R	Floor planning
16384	10	18	10	17	38	86	136	41	120	400
8192	12	18	11	14	36	67	68	20	120	400
4096	14	18	12	11	33	29	34	10	120	400
2048	4	18	12	10	30	17	17	5	120	400
1024	18	18	9	14	27	22	8	2	120	400

Tabell 8.8 Roke Manor Research FFT-transformasjon ved forskjellig konfigurasjon og optimalisering, ill. fra [32]

Ytelsestallene viser at denne transformasjonen prosesserer ett datasample pr klokkesyklus som de fleste andre høyhastighets transformasjoner, men denne kjernen skal ved optimalisert plassering i FPGA-kretsen kunne kjøres med klokkefrekvens på opptil 400 MHz. Da vil transformasjonstider på 41 μ s for en 16384-punkters FFT oppnås. Altså en meget rask FFT-transformasjon, men i modulatorimplementasjonen er pipelineforsinkelsen vel så avgjørende. Denne er imidlertid ikke oppgitt i det enkle databladet fra Roke Manor Research.

8.5.4 RF Engines Limited – Vectis QuadSpeed

RF Engines limited (RFEL), ref [33], er en hyllevare IP-kjerne leverandør og utviklingsfirma for front-end RF signalbehandlingsløsninger. De leverer IP-kjerner i et bredt spekter av anvendelser for både Xilinx og Altera FPGA-kretser.

Vectis QuadSpeed er en serie av høytytelse pipeline FFT-kjerner for sanntid prosessering av komplekse data med datarate opp til 400 MSPS. Det hevdes at denne dataraten kan økes til 800 MSPS ved bruk av deres *Distributed Half-Band Filter* (DHBF) IP-kjerne. I denne avhandlingen er det imidlertid ikke foretatt noen videre undersøkelse av dette forhold.

Ved gjennomgang av QuadSpeed datablad, ref [33], er Tabell 8.9 ført opp med pipelineforsinkelser beregnet for noen aktuelle FFT-transformasjonsstørrelser ved 400 MHz klokke rate. Tabell 8.9 viser at Vectis QuadSpeed er en rask IP-kjerne for FFT-transformasjon. På grunn av pipelinestrukturen vil utgangsdata strømme igjennom modulatorens FCP-konvolusjoner mye hurtigere enn om transformasjonene var av typen les inn data, prosesser data og les ut data.

QuadSpeed kjernen er tilgjengelig for FFT-størrelser i området 8 – 131072 datapunkter og kan spesialbestilles for større størrelser og eventuelt for variable størrelser. Databussbredder for inngangsdata, koeffisientene og utgangsdata kan bestilles innen et stort område, samt skaleringsfaktorene mellom de enkelte stegene. Organisering og reversering av inngangsdata og utgangsdata må på samme måte spesifiseres ved bestilling.

FFT-størrelse		Forsinkelse		
antall datapunkter	uten organisering av inngangsdata	med organisering av inngangsdata		
(N)	(klokkesyklar)	(μ s)	(klokkesyklar)	(μ s)
1024	1077	2,69	1332	3,33
8192	8285	20,71	10332	25,83
16384	16421	41,05	20516	51,29
32768	34877	87,19	43068	107,67
65536	65269	163,17	81652	204,13
131072	130685	326,71	163452	408,63

Tabell 8.9 Vectis QuadSpeed FFT-transformasjon, beregnet innskuddsforsinkelse

I utgangspunktet lisensieres disse IP-kjernene pr stykk for en enkelt konfigurasjon. For modulatorene i EKKO II vil det være avgjørende at antall datapunkter i IP-kjernen kan velges av brukeren, slik som IP-kjernene fra Xilinx CoreGen. Det er også litt problematisk at IP-kjernens konfigurasjonen vedrørende organisering av inngangsdata må bestemmes fra fabrikk. En enkel FCP-konvolusjon bestående av 3 like store FFT/IFFT-transformasjoner vil dermed kreve 2 forskjellige IP-kjerner på grunn av forskjellig format på inn og utdata.

8.6 Konklusjon

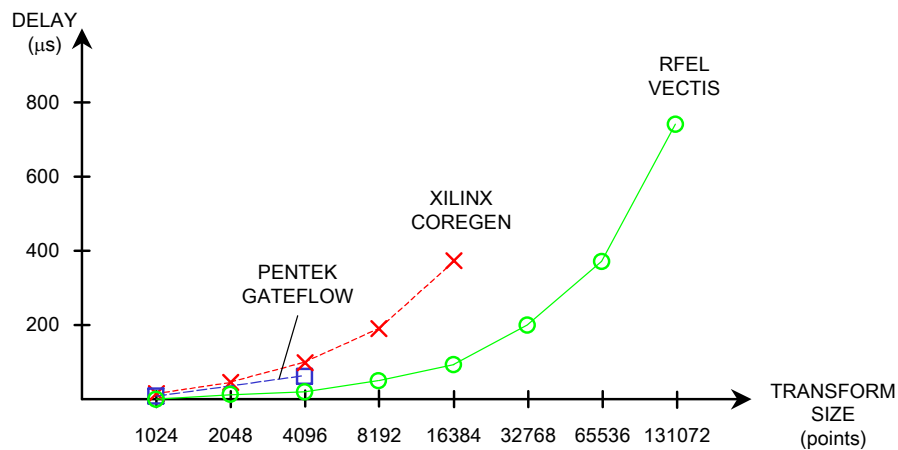
Denne analysen har sammenlignet tilgjengelige FPGA-kretser og har foreslått en komponent for modulatorene som er egnet for tyngre digital signalbehandling og som har god I/O-kapasitet med et stort antall generelle I/O-pinner og serielle høyhastighets transceivere. Kretsen XC2VPX70 fra Xilinx Virtex-II Pro X familien har en mengde dedikerte multiplikatorer og hukommelsesområder som er spesielt anvendelige ved signalbehandling.

Det er foreslått en FPGA-struktur for modulatorene som ivaretar ønskene om å kunne implementere både direkte realisert og beregnet realisert modellering. Denne strukturen er dokumentert og vist for 6 FPGA-kretser av samme type. Dette er det maksimale antall kretser som antas å kunne plasseres på et modulatorekort i EKKO II. Det foreslås også at det monteres RAM-hukommelse i tillegg til hver FPGA-krets.

Implementasjonseksemplene med direkte realisert og beregnet realisert modellering viser klart hvor mange flere reflektorer som kan modelleres ved beregnet realisert enn direkte realisert modellering med en gitt mengde ressurser. Forholdstallet fra eksemplet er ca 1700 ganger. Ulempene med beregnet realisering er modulatorens innskuddsforsinkelse. I en del

tilfeller fører dette til at modulatorens avstandsområde blir begrenset for å holde kravene til forsinkelse og maskinvarekompleksitet nede. En kombinasjon av noen modulatkort konfigurert med direkte realisering og noen modulatkort med beregnet realisering vil bedre situasjonen i en del tilfeller.

Siste del av analysen har vurdert tilgjengelige IP-kjerner for FFT-transformasjonene. Det fastslås at det er begrenset utvalg i egnede IP-kjerner. Figur 8.5 viser beregnede data for innskuddsforsinkelse gjennom modulatorene med beregnet realisert modellering og forskjellige FCP-konvolusjonsstørrelser. Kurvene i figuren representerer henholdsvis Xilinx CoreGen, Pentek Gateflow og RFELs Vectis IP-kjerner. Roke Manor Research IP-kjernen hadde mangelfullt datablad og ble derfor ikke med i sammenligningen.



Figur 8.5 Modulator innskuddsforsinkelse ved beregnet realisering og forskjellige FCP-konvolusjonsstørrelser

RFELs IP-kjerne er den eneste som er tilgjengelig i både små og store transformstørrelser, samtidig som den er raskest av alle som ble sammenlignet. Pentek Gateflow har moderat ytelse, men finnes kun i to små størrelser. Xilinx CoreGen er anvendelig så lenge transformasjonene ikke blir lengre enn 16384 datapunkter og det er moderate ytelseskrav.

Dette fører til følgende anbefaling: I applikasjoner med moderate krav til ytelse vil en standard IP-kjerne levert sammen med FPGA-utviklingsprogramvaren ofte være en god løsning, men ved krav til svært høy ytelse må en pipeline IP-kjerne fra et spesialfirma anskaffes. RFELs Vectis QuadSpeed vil være best egnet for modulatkortene.

9. Konklusjon og videre arbeider

Denne hovedfagsavhandlingen har behandlet syntetiske modelleringer av radarrefleksjoner, foreslått strukturer og implementasjoner av disse i FPGA for radarjammeren EKKO IIs modulatkort.

Avhandlingen inneholder innledningsvis noen kapitler om radarens prinsipielle virkemåte og dens historie, samt et teknologikapittel om DRFM som er en sentral systemkomponent for dagens jammesystemer. Deretter gjennomgås forskjellige metoder for modellering av radarrefleksjoner og en datamodell som er spesielt utviklet for dette formålet.

Videre omhandles tre sentrale tema i forbindelse med syntetisk modellering og DRFM konstruksjon:

Første tema er analyse av forskjellige DRFM-modulatorstrukturer for syntetisk modellering. Disse strukturene har blitt sammenlignet mot dagens direkte realiserede modellering i EKKO II jammeren. Det er blitt utviklet strukturer både med høyt og lavt ambisjonsnivå, og med forskjellig kompleksitet. Avhengig av gitte operative situasjoner og teknologiske begrensninger vil ønsket struktur som konfigureres i modulatorene variere. Dette taler til fordel for FPGA-teknologiens fortrinn – hurtig omprogrammering fra situasjon til situasjon.

Digitaliseringen er et annet sentralt tema ved DRFM-konstruksjon. Det blir analysert hvor mange bits oppløsning som er tilstrekkelig ved mottak av radarsignaler og hvor stor oppløsning det utsendte jammesignalet må ha for å oppføre seg som troverdige radarrefleksjoner etter mottak og prosessering i et høyoppløselig radarsystem.

Siste tema i avhandlingen tar for seg mulige modulatorimplementasjoner i FPGA for EKKO II jammeren. Her blir en FPGA som er egnet for tyngre digital signalbehandling foreslått, sammen med en koblingsstruktur for disse FPGA-kretsene på et modulatkort, slik at svært store konstruksjoner kan implementeres. I disse konstruksjonene er valget av IP-kjerner for FFT-transformasjon et viktig punkt. Det blir identifisert tilgjengelige IP-kjerner både for moderate ytelseskrav og for svært høye krav.

I hovedfagsarbeidet har utviklingen av forskjellige DRFM-modulatorstrukturer vært rettet mot at radarens bølgeform har hatt relativ kort varighet i tid, og derav kort utbredelse i avstand, mens radarscenen som skulle dekkes var lang i avstand. Dette er ikke alltid tilfelle, da situasjoner finnes hvor radarens bølgeform både har tilsvarende og endog lengre utbredelse i avstand enn den delen av radarscenen som ønsker jammedekning.

Det er en naturlig videreføring av arbeidene å tilpasse de foreslåtte strukturene til en slik situasjon. Til nå har samplene av radarens bølgeform vært bestemmende for konvolusjonens størrelse, og radarscenen har blitt delt opp i flere parallelle grener. I videreføringen bør det utvikles en struktur hvor både radarens bølgeform og radarscenen deles i forskjellige grener.

Et annet punkt som bør utredes er om forsinkelsesblokkene mellom de forskjellige grenene i parallelle / distribuerte FCP-konvolusjoner kan velges slik i verdi at i praksis er det kun to grener som prosesserer samplene samtidig. I så tilfelle er de resterende grenene redundante, da gjenbruk av de to første vil være mulig.

Ved å kombinere de to videreføringsforslagene kan en 2-dimensjonal struktur lages, med to parallelle grener i hver dimensjon. Den ene dimensjonen håndterer radarens bølgeform, mens den andre håndterer radarscenen. Denne strukturen vil være stor og kompleks, men vil være i stand til å håndtere 'alle' situasjoner.

Når arbeidene som er foreslått i avhandlingen og eventuelle videreføringer er implementert, vil det være naturlig å se på modellering av radarrefleksjoner for 2-D avbildende radarer som SAR og ISAR. Grunnprinsippene er mye de samme som for modellering av HRR-radar, som er omhandlet i denne avhandlingen, og for 2-D radarer, men 2-D radarer vil trenge et mer sofistikert og hurtigere oppdatert modulasjonssignal. De enkelte reflektorene i 2-D scenen må inneholde riktige dopplerfaser for hver utsendte bølgeform, i tillegg til hva som trengs for HRR-radar.

Prosesseringen av hvilken dopplerfase hver enkelt reflektor i scenen skal ha over tid, vil kunne sammenlignes med deler av SAR-prosesseringen, men bare i motsatt rekkefølge. For å utføre SAR-jamming korrekt vil det være avgjørende å vite SAR-radarens posisjon eksakt gjennom belysningstiden. Dette kan antakeligvis ivaretas med en interferometrisk mottaker i jammeren. Hvor mye av prosesseringen som kan gjøres i FPGA og hvor mye som vil bli gjort 'off-line' gjenstår å se.

Kildeliste

- [1] Saper R. H. and Dyck D. M. "A computed approach to ECM for deception of high resolution radar", Proceedings from 5th International conference on radar systems, Brest, France, May 1999
- [2] Høydal T.O. "Advanced Digital Radio Frequency Memory (DRFM) Technology – New Capabilities for Intelligent Radar Electronic Countermeasures", ATEDS/SA Symposium and Exhibition, San Diego, USA, March 2001
- [3] Kristoffersen S. and Thingsrud Ø. "The EKKO II Synthetic Target Generator for Imaging Radar", Proceedings from 5th European Conference on Synthetic Aperture Radar, Ulm, Germany, May 2004
- [4] Pace P. E. , Fouts D. J. , Ekestorm S. and Karow C. "Digital false-target image synthesiser for countering ISAR", IEE Proc.-Radar Sonar Navig., Vol 149, No 5, October 2002
- [5] He S. and Torkelson M. "A New Approach to Pipeline FFT Processor", Proceedings from 10th International Parallel Processing Symposium, Honolulu, Hawaii, April 1996
- [6] Levanon N. (1988) "Radar Measurements", Wiley Inter-Science
- [7] Wehner D. R. (1995) "High-Resolution Radar", 2nd edition, Artech House
- [8] Barton D. K. (1988) "Modern Radar Systems Analysis", Artech House
- [9] Skolnik M. I. (1990) "Radar Handbook", 2nd edition, McGraw-Hill
- [10] Oliver C. and Quegan S. (1998) "Understanding Synthetic Aperture Radar Images", Artech House
- [11] Stearns S. D. and Hush D. R. (1990) "Digital Signal Analysis", Prentice Hall
- [12] Schleher D. C. (1986) "Introduction to Electronic Warfare", Artech House
- [13] Pace P. E. (2000) "Advanced Techniques for Digital Receivers", Artech House
- [14] Sala L. (1990) "Digital Radio Frequency Memory (DRFM) Implementation of Velocity Deception", Anaren Microwave Inc
- [15] Anaren Microwave Inc (1994) "Technical Manual, Digital RF Memory model 450170", Anaren Microwave Inc
- [16] Xcell Journal, Issue 48, Spring 2004
- [17] Scientific biography library – James Clerk Maxwell, <http://scienceworld.wolfram.com/biography/Maxwell.html>, Juli 2004

- [18] Scientific biography library – Heinrich Hertz, <http://scienceworld.wolfram.com/biography/HertzHeinrich.html>, Juli 2004
- [19] Scientific biography library – Sir Edward Victor Appleton, <http://scienceworld.wolfram.com/biography/Appleton.html>, Juli 2004
- [20] IEEE History Center – Lawrence A. Hyland, http://www.ieee.org/organizations/history_center/legacies/hyland.html, Juli 2004
- [21] Radar world web site, <http://www.radarworld.org>, Webside for radarhistorie, Juli 2004
- [22] Aircraft technology web site, <http://www.airforce-technology.com/projects/fl17>, Juli 2004
- [23] EUSAR 2004, <http://213.144.9.81/veranstaltungen/de/va/-eusar2004.htm>, Konferansens hjemmeside, Juli 2004
- [24] Canadian Space Agency, <http://radarsat.space.gc.ca/asc/eng>, Firmaets hjemmeside, Juli 2004
- [25] Analog Devices, <http://www.analog-devices.com>, Firmaets hjemmeside, Juli 2004
- [26] Maxim Integrated Products, <http://www.maxim-ic.com>, Firmaets hjemmeside, Juli 2004
- [27] Integrated Device Technology, <http://www.idt.com>, Firmaets hjemmeside, Juli 2004
- [28] Xilinx Inc, <http://www.xilinx.com>, Firmaets hjemmeside, Juli 2004
- [29] Triquint Semiconductor, <http://www.triquint.com>, Firmaets hjemmeside, Juli 2004
- [30] Texas Instruments, <http://www.ti.com>, Firmaets hjemmeside, Juli 2004
- [31] Pentek Inc, <http://www.pentek.com>, Firmaets hjemmeside, Juli 2004
- [32] Roke Manor Research, <http://www.roke.co.uk>, Firmaets hjemmeside, Juli 2004
- [33] RF Engines Limited, <http://www.rfel.com>, Firmaets hjemmeside, Juli 2004