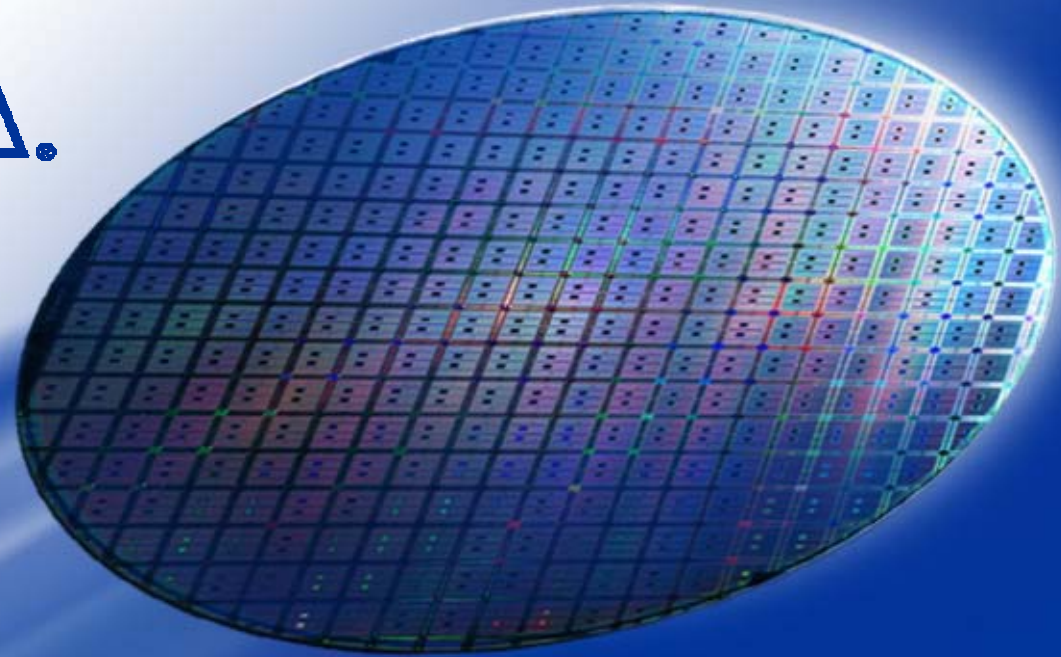


ALTERA.



Reconfigurable Radio with FPGA-Based Application-Specific Processors

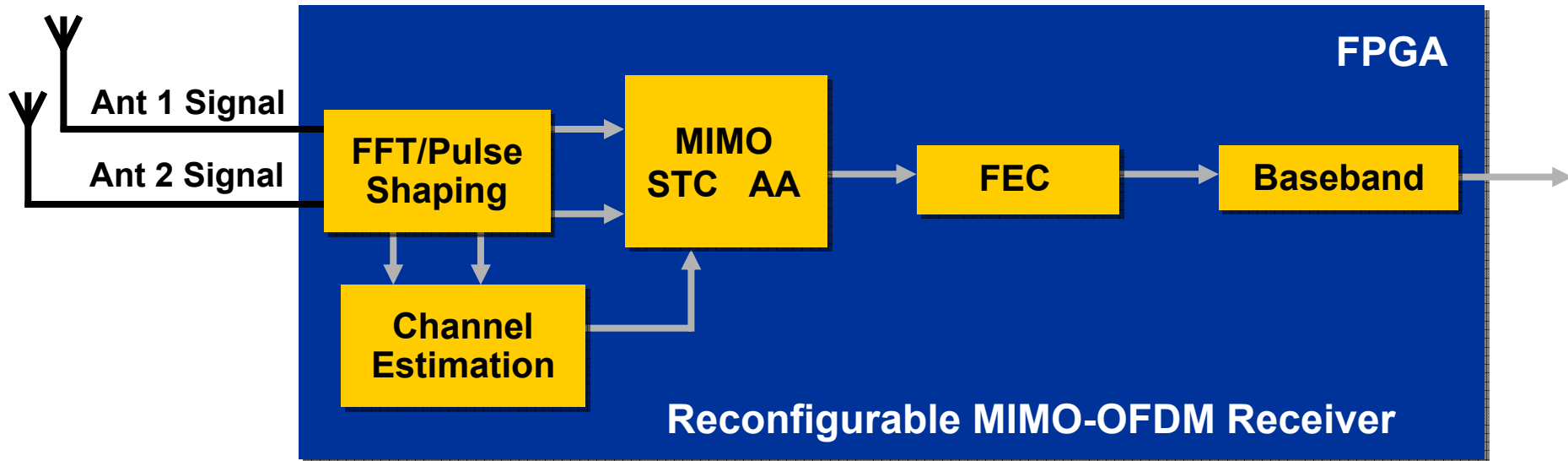
Rob Jackson Sam Hettiaratchi Mike Fitton Steve Perry

Altera European Technology Centre

Agenda

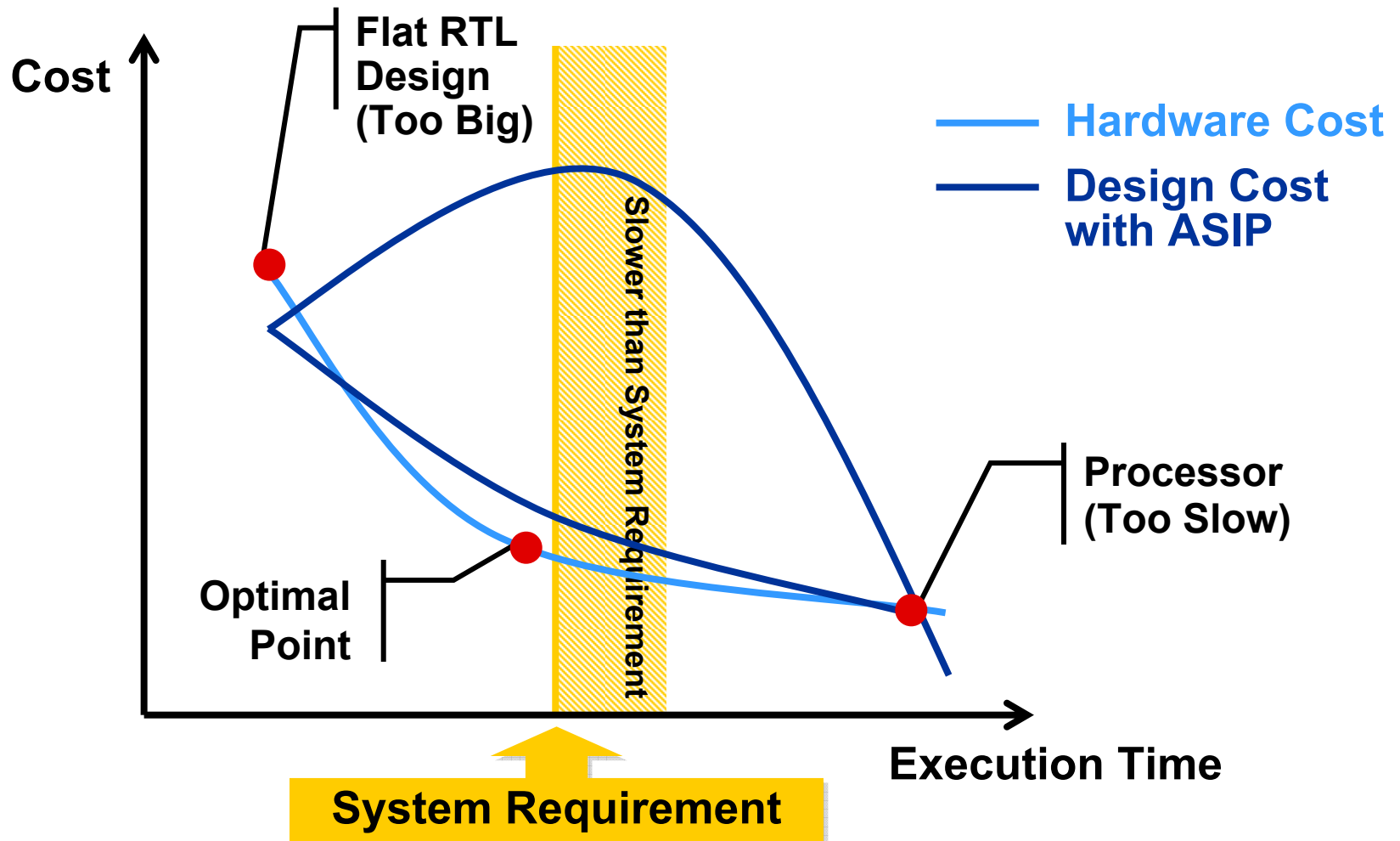
- Motivation
 - System-Level Design
 - Reaching Optimal Cost/Performance Point
 - Productivity
- Application Specific Integrated Processors (ASIPs)
- ASIPs in Software Defined Radio
 - Programmability & Reconfiguration
 - An Example: FIR/FFT Implementation
- Summary

Motivation: System Concept



- Implement Algorithms Quickly & Efficiently
- Ensure Desired Cost/Performance Trade-Off
- Usability: Present a Software-Like Interface
- Tight Integration with Tools & System on a Programmable Chip Concept

Accessing the Optimal Point



Productivity Motivation

- *Design Effort* is High for FPGAs Compared to other Programmable Platforms

BDTi

Grading the Alternatives

	DSPs	GPps	FPGAs	Custom Cores	ASICs	ASSPs
Design Effort	B	A	D	C	E	A+
Design Flexibility	E	E	B	C	A	E
Run-time Flexibility	C	B	A	C	E	E
Top Speed	D	E	B	C	A	A
Energy Efficiency	C	D	C	B	A	A

A = Best, E = Worst

© 2003 Berkeley Design Technology, Inc. 34

Agenda

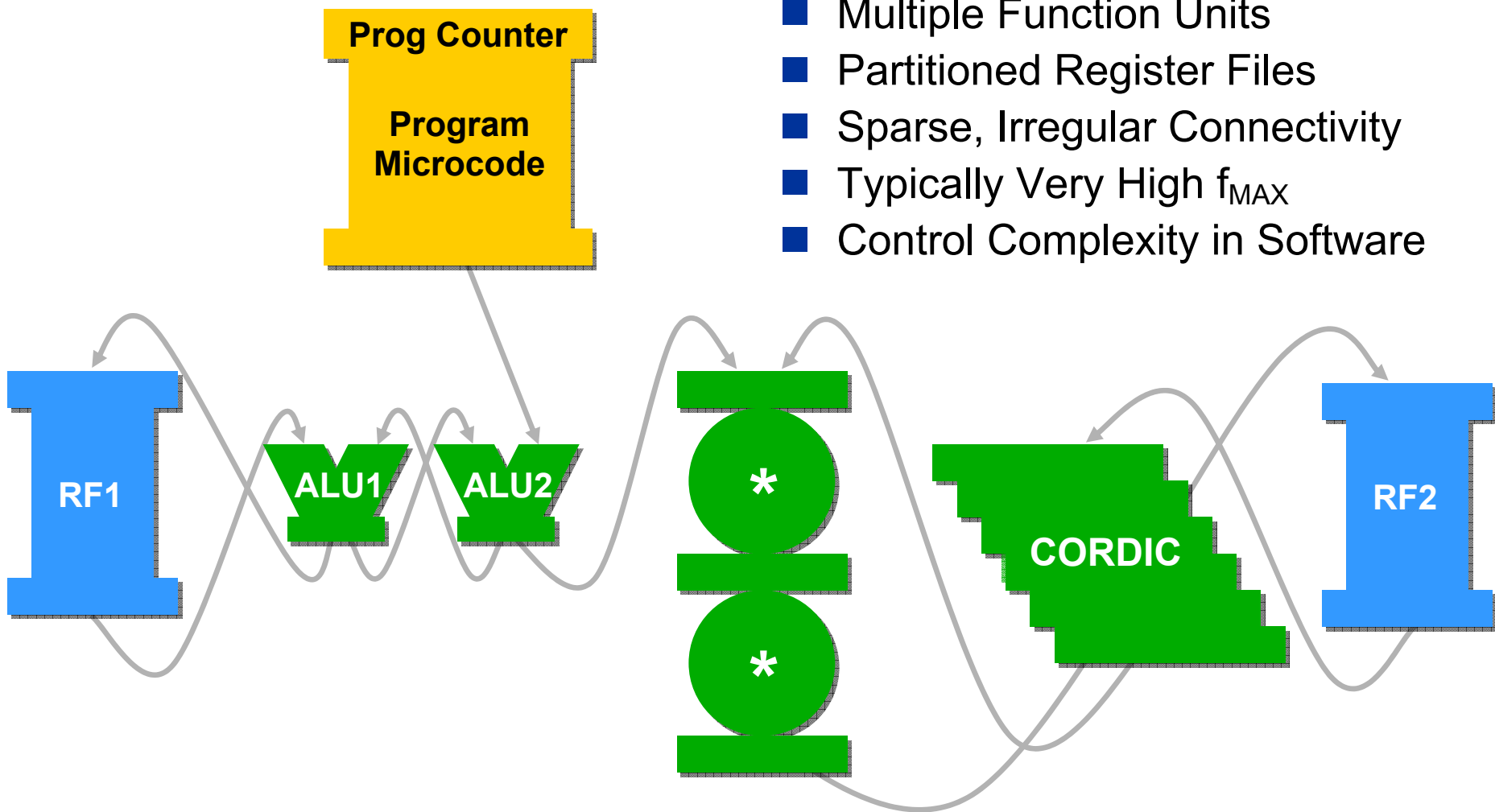
- Motivation
 - System-Level Design
 - Reaching Optimal Cost/Performance Point
 - Productivity
- Application Specific Integrated Processors (ASIPs)
- ASIPs in Software Defined Radio
 - Programmability & Reconfiguration
 - An Example: FIR/FFT Implementation
- Summary

ASIP Approach

- Build Application-Specific Processors
 - Easy to Think of Them as Custom Processors
- Application-Specific Integrated Processor Has
 - Flexible Number of Functional Units
 - Function Units Connections are Defined by the Algorithm
 - Program to Allow Flexible (& Dynamic) Control
 - Software-Based Design Flow

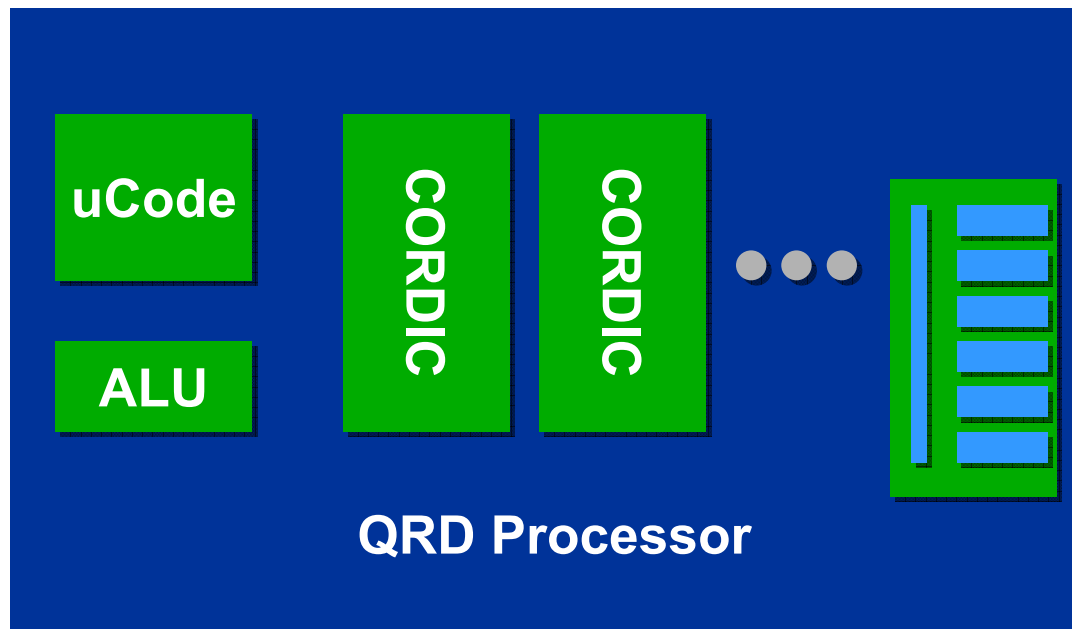
***Software Process brings
Productivity vs. RTL***

Example ASIP



Hierarchical Composition

- Processors That Are Built Can Be Used As Functional Units Within Another Processor
- Users Can Add Functional Units to Their Processors



Agenda

- Motivation
 - System Level Design
 - Reaching Optimal Cost/Performance Point
 - Productivity
- Application Specific Integrated Processors (ASIPs)
- ASIPs in Software Defined Radio
 - Programmability & Reconfiguration
 - An Example: FIR/FFT Implementation
- Summary

ASIPs in SDR: Programmability

- The ASIP Application is a Software Program
 - Sequential High Level Code
 - Instruction Level Parallelism
- RTL is a Hardware Description Language
 - Behavior Encoded in a Number of Parallel Processes
- Software Can be:
 - Ported, Adapted, Updated, Enhanced
- ASIPs Can Combine the Flexibility of DSPs With the Processing Power of FPGAs

ASIPs in SDR: Reconfiguration

- ASIP Uses a Program
 - Can be Quickly & Easily Reprogrammed by Changing the Contents of the Program Memory
- Conventional Hardware Description
 - Reconfiguration Requires a New FPGA Image - Literally Rewiring the Device
- Migrate to Structured ASIC (e.g. Altera Hardcopy Structured ASIC)
 - Does Not Compromise the Reconfigurability

Methodology

1. Analyse Requirements
2. REPEAT{
3. Select & Parameterise Function Units, Choose Topology
4. Assess Implementation Performance
5. } UNTIL Solution Found

■ FFT Algorithm Requirements

- Radix-2, Complex, 1,024-Point, Decimation in Time
- 2 Data Read, 2 Data Write, 1 Coefficient Read, 6 Add/Sub, 4 Multiply Per Butterfly

■ FIR Algorithm Requirements

- Simple 10-Tap Complex FIR
- 1 Data Read, 1 Coefficient Read, 4 Add/Sub, 4 Multiply Per Tap

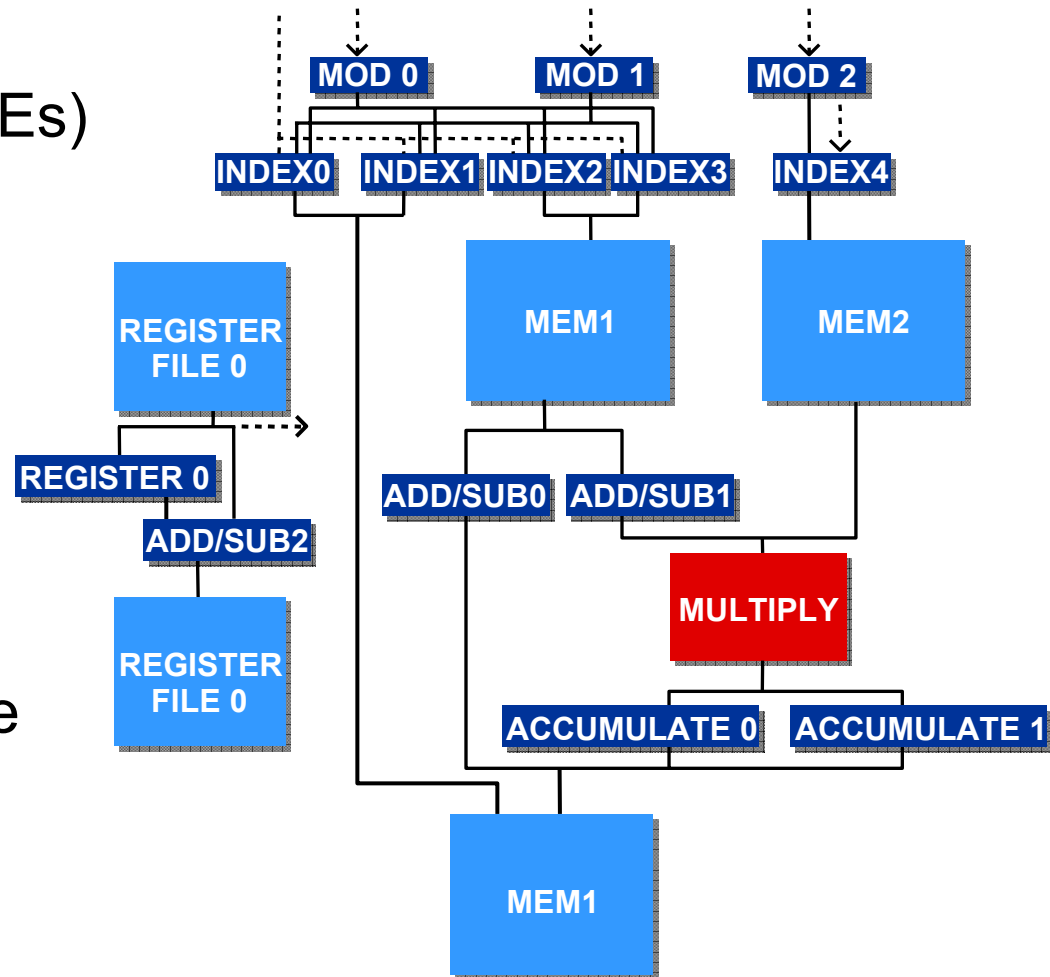
Methodology

1. Analyse Requirements
2. REPEAT{
3. Select & Parameterise Function Units, Choose Topology
4. Assess Implementation Performance
5. } UNTIL Solution Found

- System Constraint
 - 1 18x18 Multiplier
- ASIP Specification
 - Optimize for ~100% Multiplier Utilization (Efficiency)
 - 1 Dual Ported Data Memory & 1 Single Ported Coefficient Memory
 - 5 Address Generators, 4 ALU (2 Used as Accumulators)
- In Effect We have Built a Simple DSP Processor Capable of Performing FIR & FFT

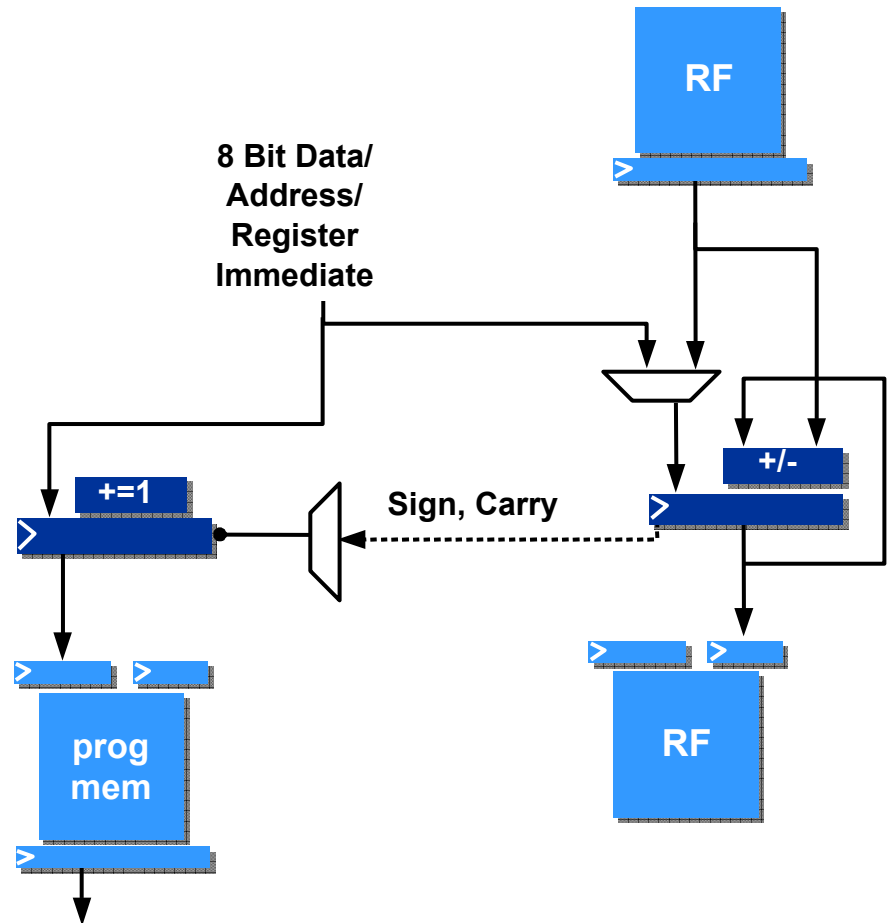
Reconfigurable FFT/FIR Filter Module

- Simple Hardware
 - 322 Logic Elements (LEs)
 - 1 18x18 Multiplier
- High Performance
 - 230 MHz in a Stratix –5 FPGA
 - 1,024 Pt FFT Takes 21,850 Cycles (95 μ s)
- Less than 5% of Available Logic in Smallest Altera Stratix FPGA



FFT/FIR Control Processor

- Complete Processor
- Can Perform Any Task
- 52 Logic Elements (LEs) & 2 M4K Memories
- Processors Implement Control Structures Very Cheaply & Densely



Program: Control

```
for (row = 0; row < number_coefficients; row++)
{
    for (col = 0; col < number_coefficients; col++)
    {
        mul = G[row] * PX[col];
        add18 = P[row][col] - mul;
        scale = add18 * scale_constant;
        P[row][col] = scale;
    }
}
```

- High Level, Symbolic Control
 - Independent of the Final Hardware Implementation

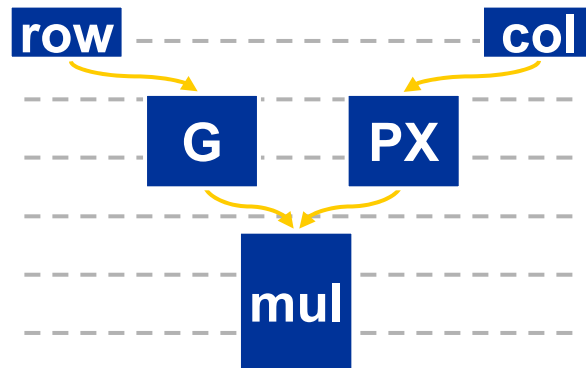
Program: Data Processing

```
for (row = 0; row < number_coefficients; row++)  
{  
    for (col = 0; col < number_coefficients; col++)  
  
    {  
        mul = G[row] * PX[col];  
        add18 = P[row][col] - mul;  
        scale = add18 * scale_constant;  
        P[row][col] = scale;  
    }  
}
```

- Statements are Sequential & ‘Well Defined’
 - No Explicit Hardware Timing

Program: One Instruction

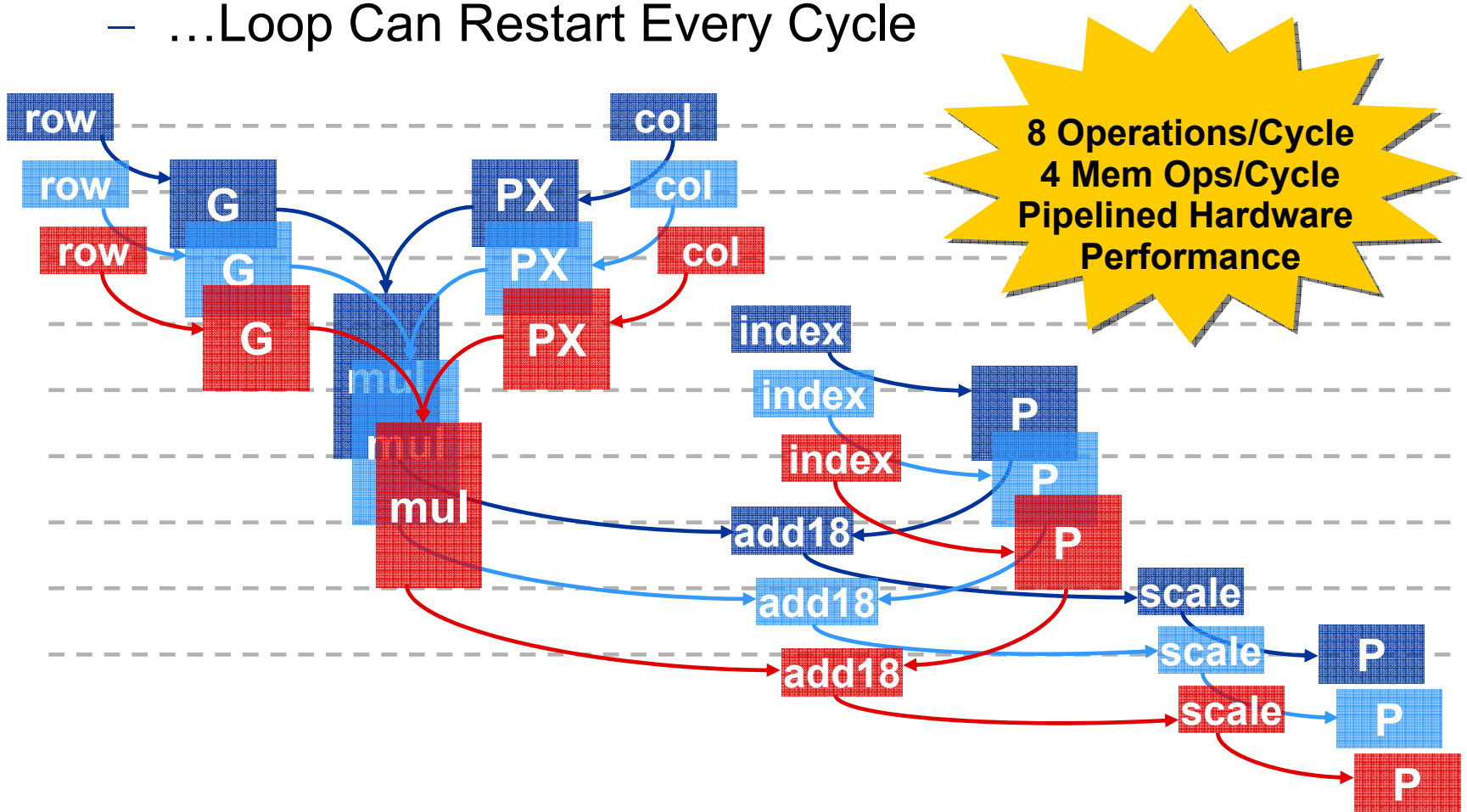
```
for (row = 0; row < number_coefficients; row++)  
{  
    for (col = 0; col < number_coefficients; col++)  
  
    {  
        mul = G[row] * PX[col];  
        add18 = P[row][col] - mul;  
        scale = add18 * scale_constant;  
        P[row][col] = scale;  
    }  
}
```



- First Instruction:
5 Cycles

Program: Pipelined Loop

- But if There Are 3 Mem, 2 Mults, 1 Adder...
 - ...Loop Can Restart Every Cycle



FFT Performance Comparison

Implementation	Size	Speed
ASIP-on-FPGA	<5% of Smallest Stratix FPGA	21,850 Cycles @ 230 MHz (95 μ s)
DSP	TI C62	20,840 Cycles @ 300 MHz (69.5 μ s)

1,024 Point Radix-2 FFT

- DSP is Faster than ASIP
- However, ASIP Only Requires Less than 5% of Stratix FPGA Device

FFT Performance Comparison

Implementation	Size	Speed
ASIP-on-FPGA	600 LEs (1 LE = One 4-Input LUT)	6,638 Cycles @ 256 MHz (Under 25 μ s)
RTL IP Core from Xilinx	1,869 Logic Slices (1 Logic Slice = Two 4-Input LUTs)	4,145 Cycles @ 100 MHz (41.5 μ s) [9]

1,024 Point Radix-4 FFT

- ASIP-on-FPGA Gives a Denser Faster Implementation

Agenda

- Motivation
 - System-Level Design
 - Reaching Optimal Cost/Performance Point
 - Productivity
- Application Specific Integrated Processors (ASIPs)
- ASIPs in Software Defined Radio
 - Programmability & Reconfiguration
 - An Example: FIR/FFT Implementation
- Summary

Summary

- Methodology for Constructing Customized Application-Specific Processors on FPGA
- Efficient Algorithm Implementations in FPGA
- FPGA Implemented ASIPs are a Good Target for SDR
 - Performance of an FPGA
 - Can be Re-Programmed Without Completing a Hardware Change Cycle
- ASIPs Remain Reprogrammable if the FPGA Design is Converted to ASIC (e.g., HardCopy Structured ASIC)