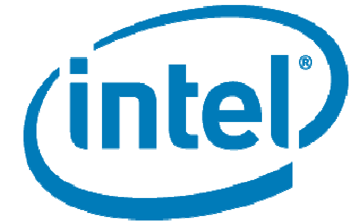


The Scalable Communications Core: A Multi-Core Wireless Baseband Prototype

Dr. Anthony (Tony) Chun
DSP Architect
Wireless Communications Lab
Corporate Technology Group
Intel Corporation
anthony.l.chun@intel.com

IEEE SCV Signal Processing Society, Feb. 9, 2009

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

Introduction

- What is the Scalable Communications Core?
  - Flexible baseband
  - Supports multiple communication standards
  - Multi-core DSP
  - Heterogeneous coarse-grained accelerators
  - NoC interconnect
- Contributions
  - Developed area and energy-efficient architecture
  - Developed programming technology
  - Taped out first test chip
  - Validated WiFi and WiMAX
  - Mapped components of Bluetooth, DVB-H and GPS

Why is this work interesting?

- Intersection of several disciplines
  - Communications
  - Signal Processing
  - Algorithms
  - Architecture
  - On chip interconnect
  - Programming tools

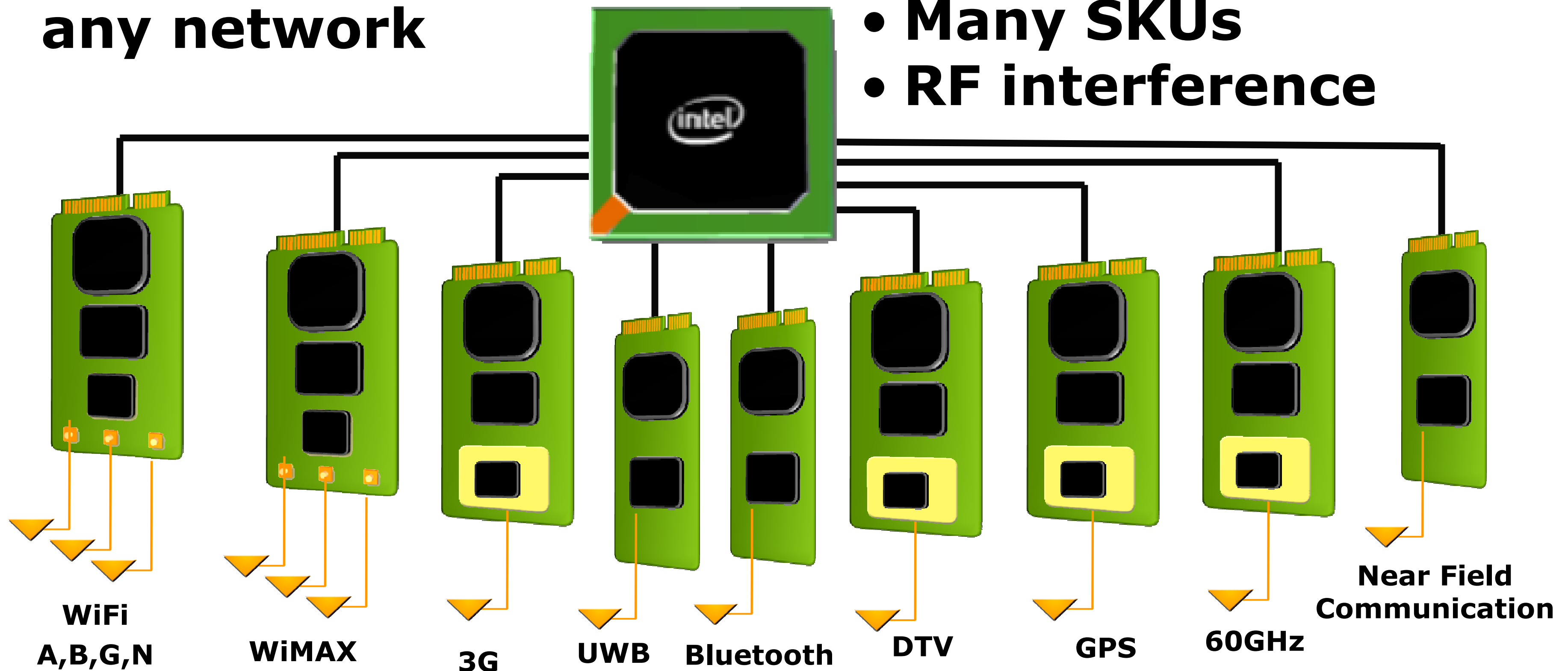
Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

Motivation: Too Many Radios in Future Platforms

Vision: Connectivity anytime, anywhere to any network

Problems:
- Large Form factor
- Many SKUs
- RF interference



intel

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

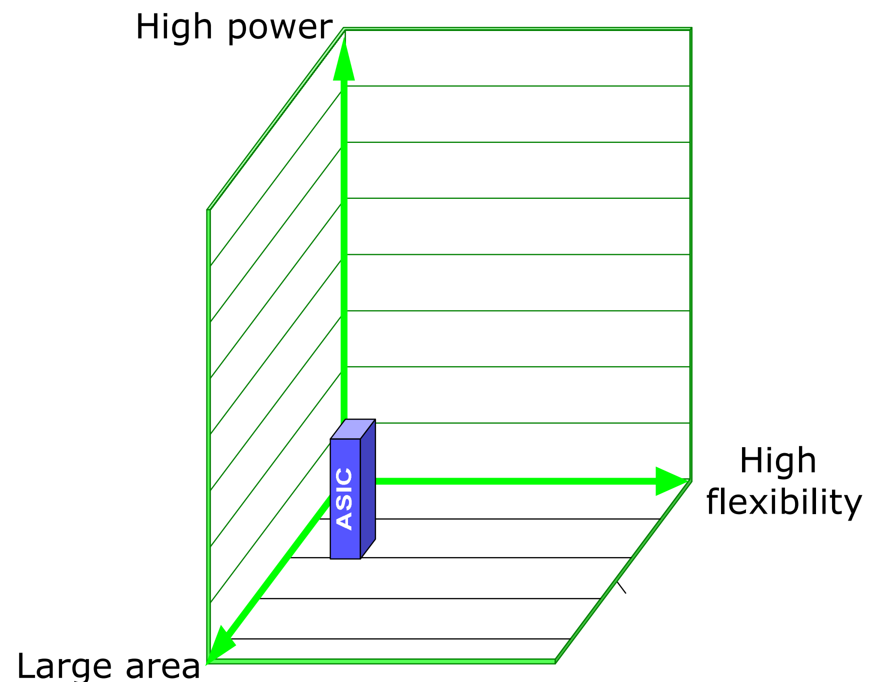
SCC Architecture Overview

- Heterogeneous coarse-grained Processing Elements
  - Each is programmable within its domain
  - Support for multiple threads within PEs
  - Stream processing
  - Distributed memory
- Network-on-Chip (NoC) interconnect
  - Packet-based
  - Direct connection to nearest neighbors
  - Stringent latency requirements
- Data-driven distributed control
  - Control embedded within packet header
  - Microcontroller is used only for low rate configuration

Flexibility Tradeoffs

- Flexible architecture trades three vectors
- ASIC: low flexibility, low power, small area
- Digital Signal Processor (DSP): high flexibility, high power, medium area
- FPGA: high flexibility, high power, large area
- SCC: medium flexibility, medium power, small area

For multiple basebands



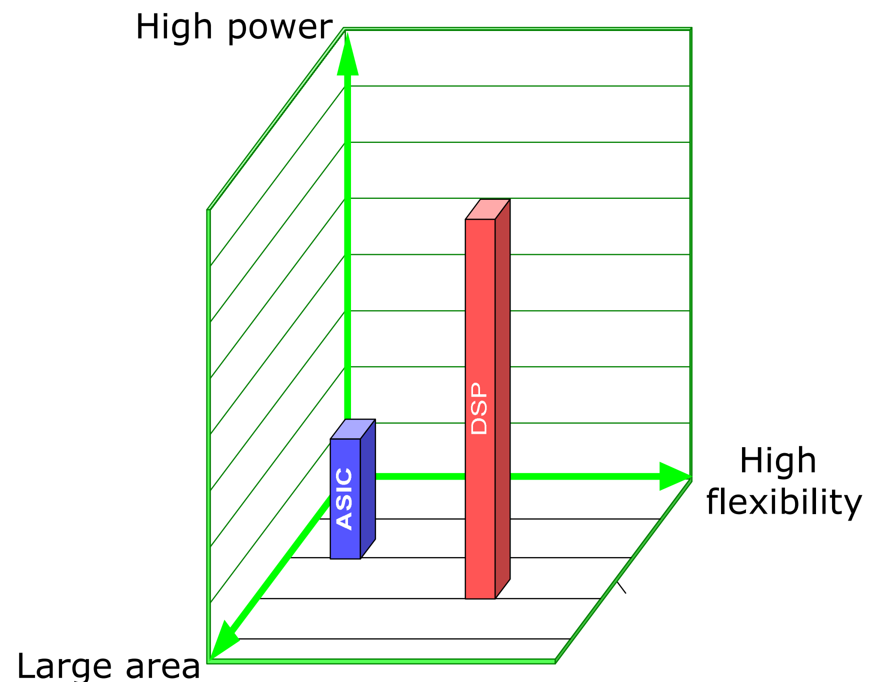
Conceptual

intel

Flexibility Tradeoffs

- Flexible architecture trades three vectors
- ASIC: low flexibility, low power, small area
- Digital Signal Processor (DSP): high flexibility, high power, medium area
- FPGA: high flexibility, high power, large area
- SCC: medium flexibility, medium power, small area

For multiple basebands

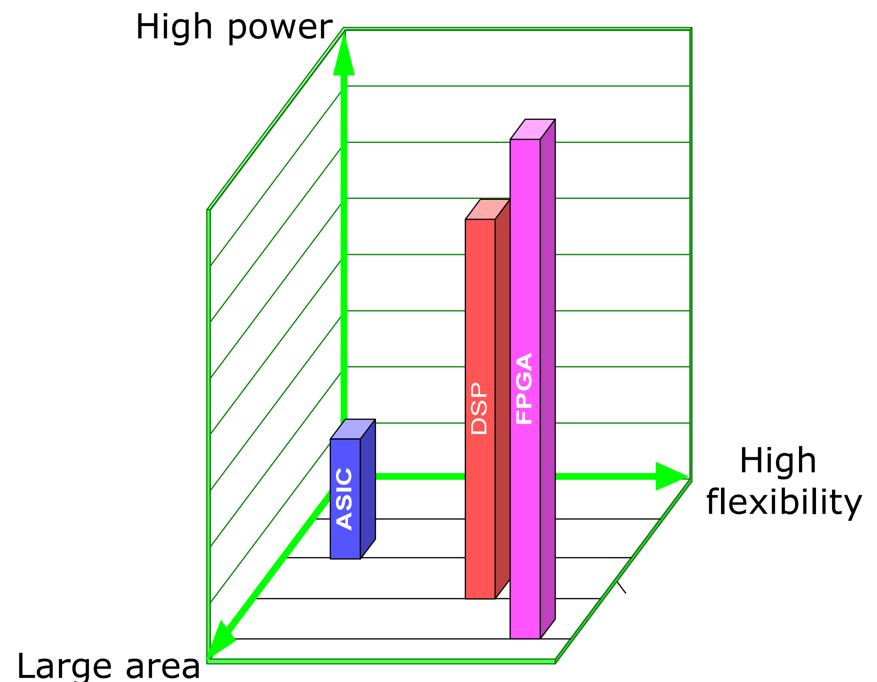


Conceptual

Flexibility Tradeoffs

- Flexible architecture trades three vectors
- ASIC: low flexibility, low power, small area
- Digital Signal Processor (DSP): high flexibility, high power, medium area
- <u>FPGA: high flexibility, high power, large area</u>
- SCC: medium flexibility, medium power, small area

For multiple basebands



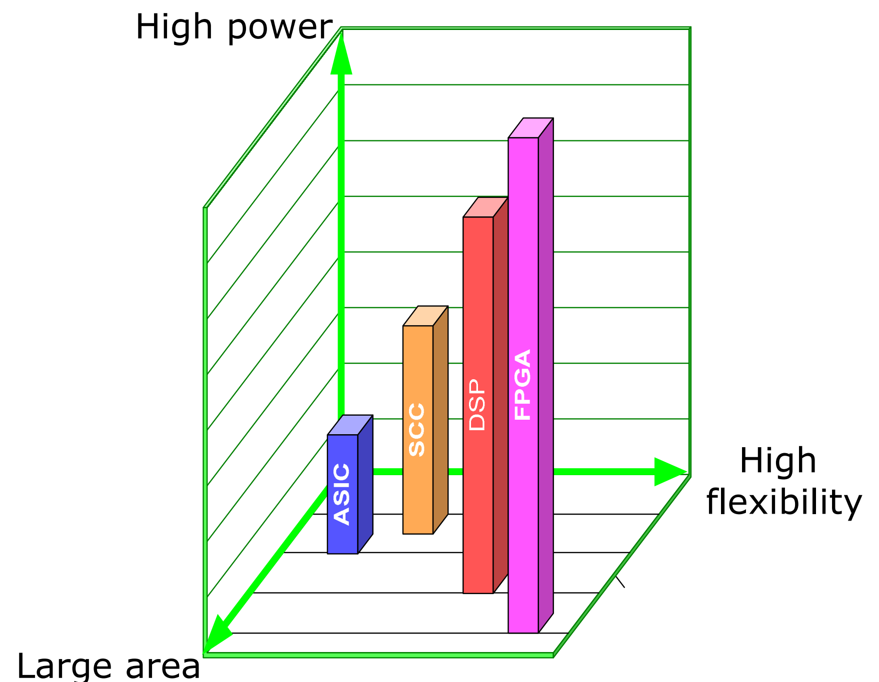
Conceptual

intel

Flexibility Tradeoffs

- Flexible architecture trades three vectors
- ASIC: low flexibility, low power, small area
- Digital Signal Processor (DSP): high flexibility, high power, medium area
- FPGA: high flexibility, high power, large area
- SCC: medium flexibility, medium power, small area

For multiple basebands



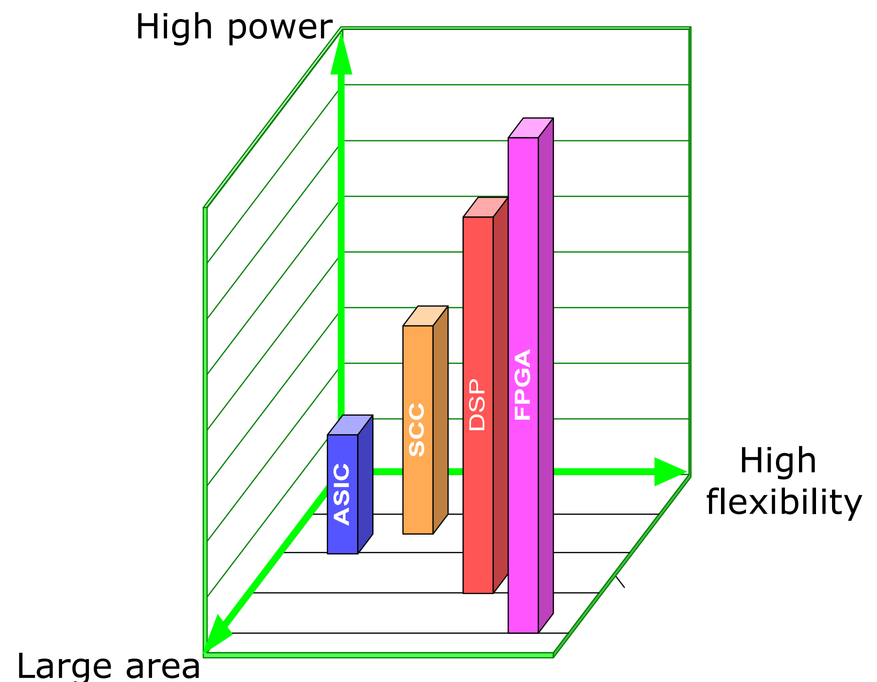
Conceptual

Flexibility Tradeoffs

- Flexible architecture trades three vectors
- ASIC: low flexibility, low power, small area
- Digital Signal Processor (DSP): high flexibility, high power, medium area
- FPGA: high flexibility, high power, large area
- SCC: medium flexibility, medium power, small area

✓ SCC solution offers best combination of energy efficiency, area efficiency and flexibility

For multiple basebands



Conceptual

Observation: Many Commonalities Between Wireless Standards

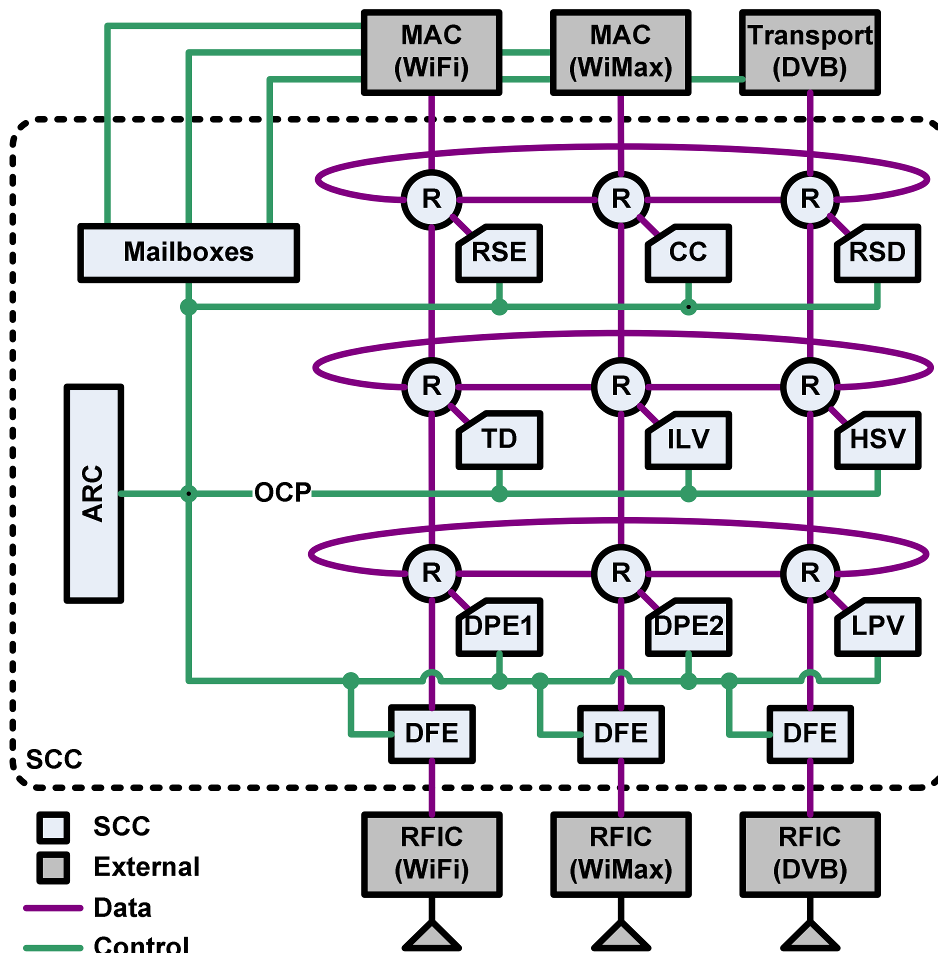
Algorithm	WiFi	WiMax	3G	DVB-T	UWB	60GHz
FIR / IIR	√	√	√	√	√	√
Correlation	√	√	√	√	√	√
Spreading			√			
FFT	√	√		√	√	√
Channel Estimation	√	√	√	√	√	√
QAM Mapping	√	√	√	√	√	√
Interleaving	√	√	√	√	√	√
Convolutional Coding	√	√	√	√	√	√
Turbo Coding		√	√			
Reed-Solomon Coding		√		√	√	√
Randomization	√	√	√	√	√	√
CRC	√	√	√		√	√

Wireless standards share many of the same DSP algorithms

Architecture Considerations

- Large superset of protocols, but only a few are active concurrently
- Complex control procedures with strict timing requirements
- Pipelined data flow through protocol stack
- Must support variable data block sizes
- Must be able to constrain timing jitter and latency

Solution: Heterogeneous Processors on a 3-ary 2-cube NoC

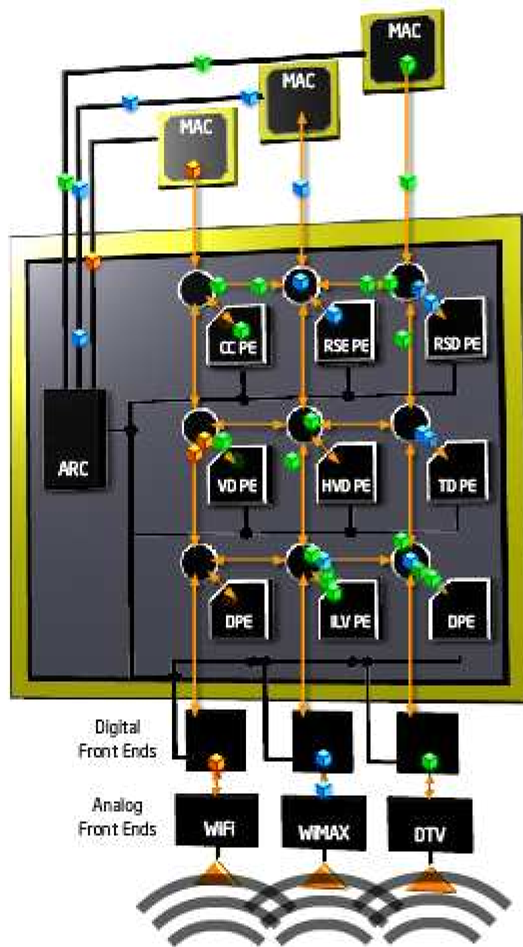


- Heterogeneous Processing Elements
  - Digital Front End (DFE)
  - Data-Stream Processing Engine (DPE)
  - Interleaving (ILV)
  - High-Speed Viterbi (HSV)
  - Low Power Viterbi (LPV)
  - Turbo-Decoder (TD)
  - Convolutional Coder (CC)
  - Reed-Solomon Decode(RSD)
  - Reed-Solomon Encode(RSE)
- 3-ary 2-cube NoC Data Plane
- 32-bit ARC™ RISC Processor
- 32-bit OCP™ Control Plane
- PLME Mailboxes

intel

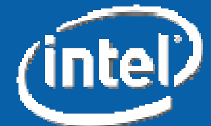
Solution: *Scalable Communications Core*

A Baseband Processor for WiFi, WiMAX, and DVB Multi-radio



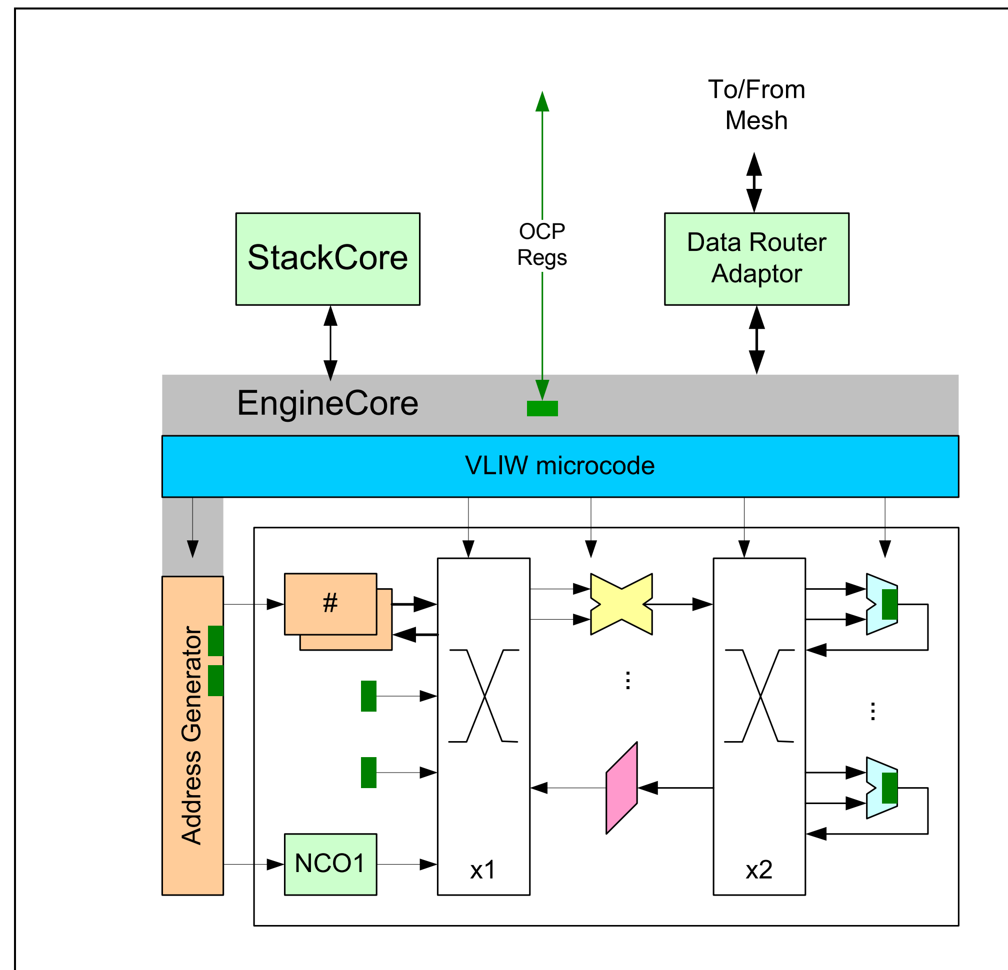
- Heterogeneous Processing Elements
  - Digital Front End (DFE)
  - Data-Stream Processing Engine (DPE)
  - Interleaving (ILV)
  - High-Speed Viterbi (HSV)
  - Low Power Viterbi (LPV)
  - Turbo-Decoder (TD)
  - Convolutional Coder (CC)
  - Reed-Solomon Decoder (RSD)
  - Reed-Solomon Encoder (RSE)
- 3-ary 2-cube NoC Data Plane
- 32-bit OCP™ Control Plane
- PLME Mailboxes
- 32-bit ARC™ RISC Processor

Heterogeneous Processing Elements on 3-ary 2-cube NoC



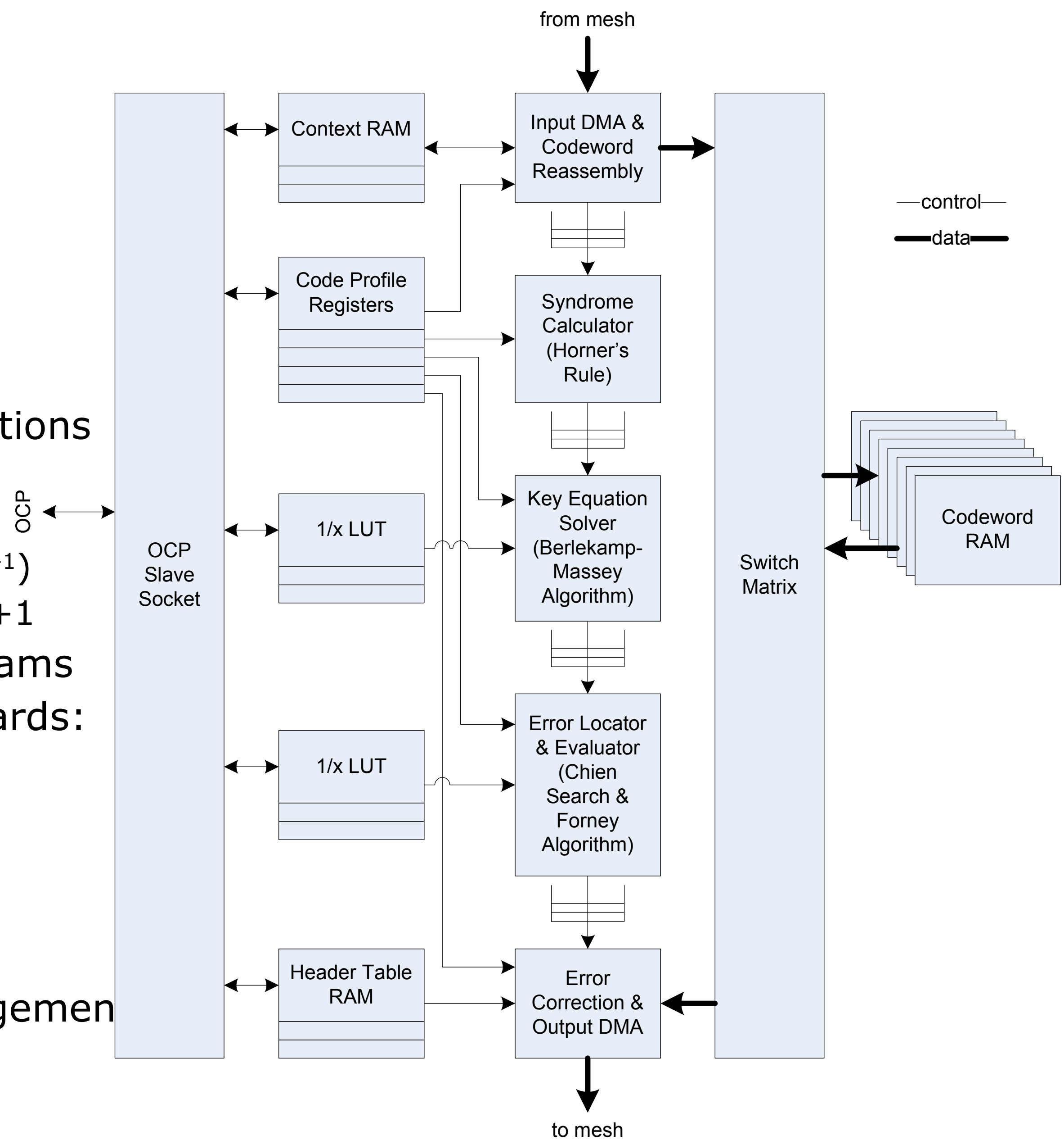
Data Stream Processing Element

- 16-bit microcontroller (StackCore)
  - Configuration
  - Micro/Macro-sequencing
  - Scalar arithmetic
  - Programmed using C or assembly
- Complex DSP machine (EngineCore)
  - Highly reconfigurable data path
  - Crossbar connections
  - Complex mult, add, sub, shift, round, sat, trunc, conj.
  - Split VLIW microcode –
    - ✓ Long Configuration Words
    - ✓ Long Address Words
  - Address Generators
  - Stream programming model



Reed-Solomon Decoding

- Maximum throughput
  - 84.2Mbps ATSC
  - 105.8Mbps DVB-H (PHY)
  - 22.9Mbps DVB-H (MPE)
- Up to 4 resident configurations
  - $GF(2^m)$; m<=8
  - T<=32
  - $g(x)=(x+1)(x+a)\ldots(x+a^{2T-1})$
  - $p(x)=c_0x^m+ c_1x^{m-1}\ldots c_{m-1}x+1$
- Up to 4 simultaneous streams
- Example supported standards:
  - ATSC
  - DVB-H
  - 802.16de
  - ITU-T J.83
- Integrated clock gating
- Fine grained power managemen



intel

Opportunity: Radio Composition using Shared Resources

- Smaller – reduce redundancy by sharing resources
- More Energy Efficient – reduced redundancy equates to lower leakage
- Scalable – can easily add new processing elements to cover emerging standards
- Wider Roaming – can compose radios on-the-fly based on signals detected in the air
- Improved Coexistence – wider array of future interference mitigation and coordination options
- Potential Time to Market Reduction – future drag and drop methodology for building a multi-radio baseband processor using well characterized processing elements on a flexible and scalable interconnect

Dataflow & Resource Sharing: WiFi vs. Mobile WiMax TX Case

MAC (WiMax)

Shared Resources

MAC (WiFi)

R R R

RSE CC RS D

- 1 Information
- na Scrambled
- na Outer Code
- na Outer Intrlv
- 2 Inner Code
- 3 Inner Intrlv
- 4 Symbols
- 5 Samples

R R R

TD ILV HSV

R R R

DPE DPE LPV

DFE DFE DFE

RFIC (WiMax)

R R R

RSE CC RS D

- 1 Information
- na Scrambled
- na Outer Code
- na Outer Intrlv
- 2 Inner Code
- 3 Inner Intrlv
- 4 Symbols
- 5 Samples

R R R

TD ILV HSV

R R R

DPE DPE LPV

DFE DFE DFE

RFIC (WiFi)

Mobile WiMAX ***WiFi***

Dataflow & Resource Sharing: Fixed WiMAX vs. DVB RX Case

Shared Resources

MAC (WiMax)

R R R
RSE CC RSD
R R R
TD ILV HSV
R R R
DPE DPE LPV
DFE DFE DFE

RFIC (WiMax)

7 Information
6 Scrambled
5 Outer Code
na Outer Intrlv
4 Inner Code
3 Inner Intrlv
2 Symbols
1 Samples

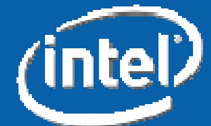
MPEG-TS (DVB)

R R R
RSE CC RSD
R R R
TD ILV HSV
R R R
DPE DPE LPV
DFE DFE DFE

RFIC (DVB)

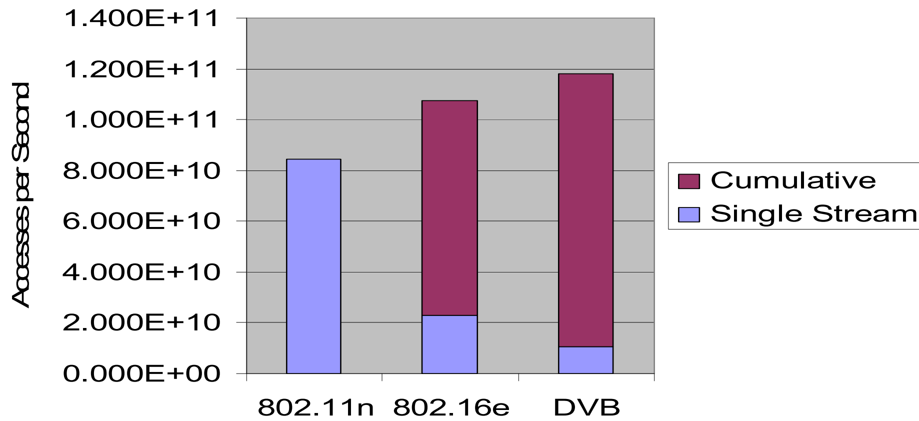
8 Information
7 Scrambled
6 Outer Code
5 Outer Intrlv
4 Inner Code
3 Inner Intrlv
2 OFDM Symbol
1 Samples

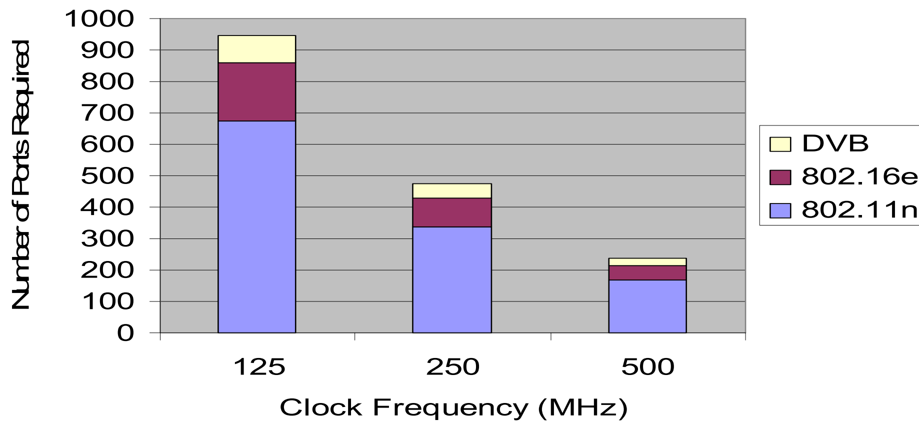
Fixed WiMAX ***DVB***



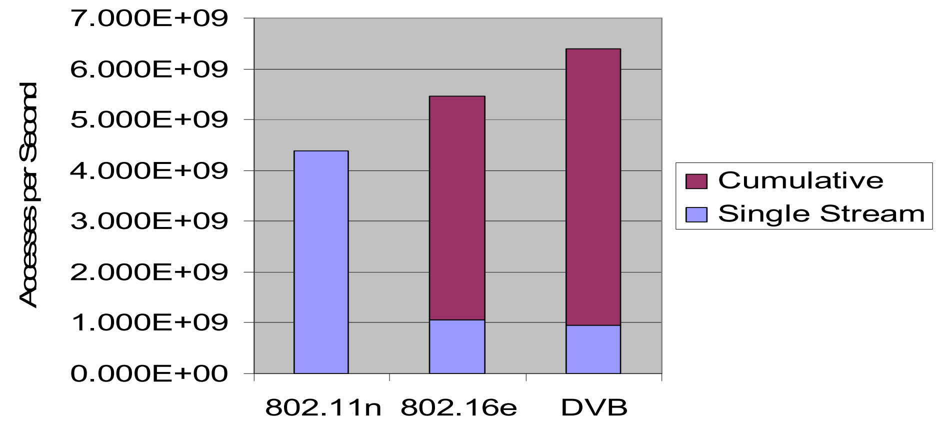
Distributed Memory

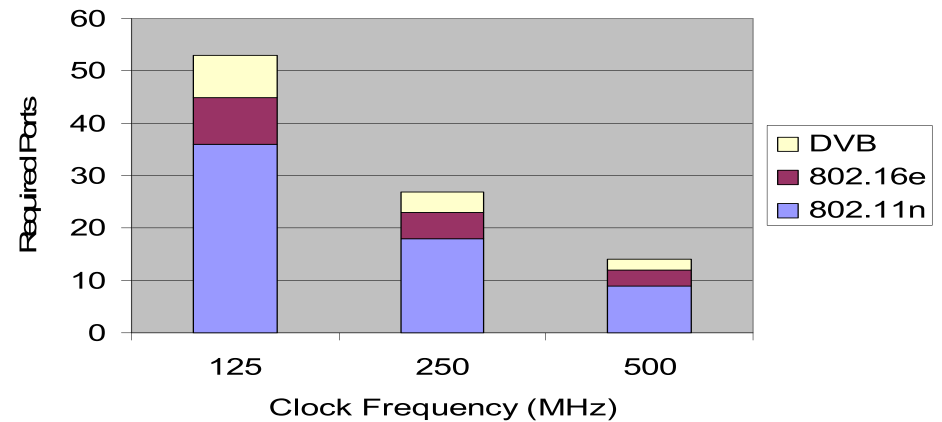
DSP + FEC



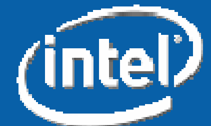


DSP alone

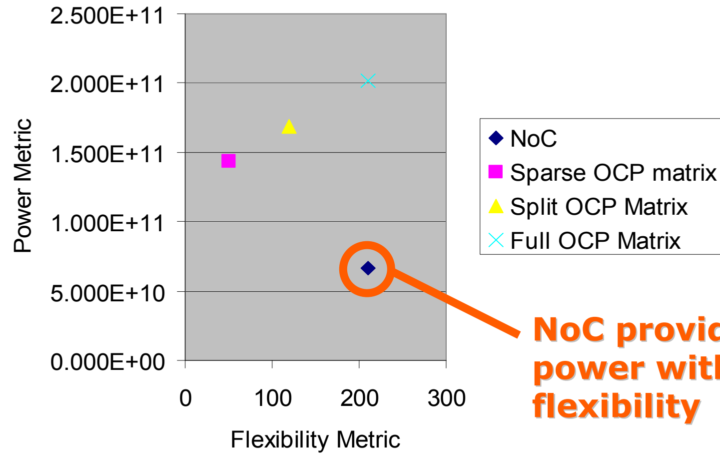




Shared memory not practical – distributed memory required for bandwidth.

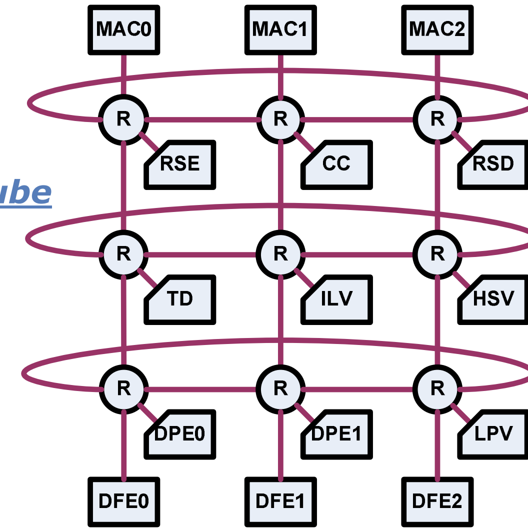


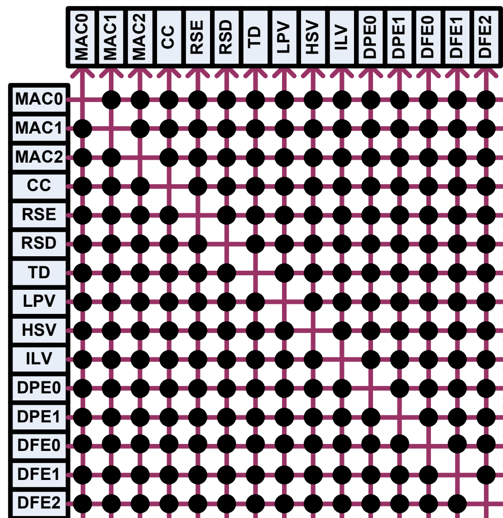
Interconnect Considerations

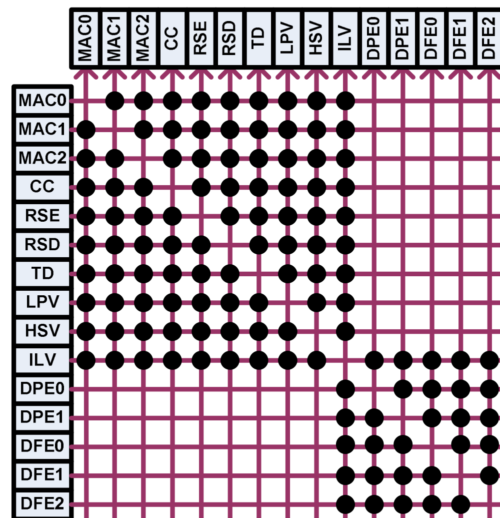


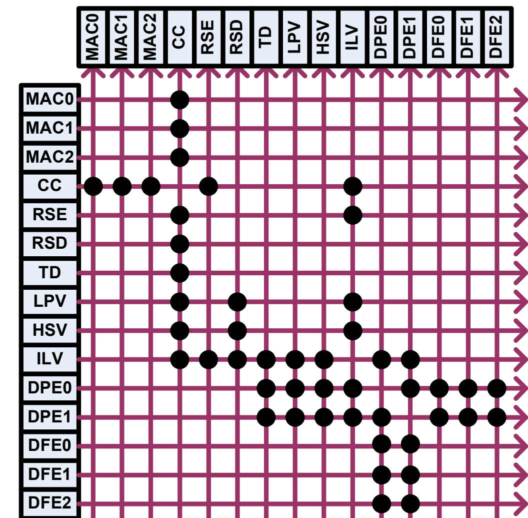
NoC provides lowest power with maximum flexibility

3-ary 2-cube NoC







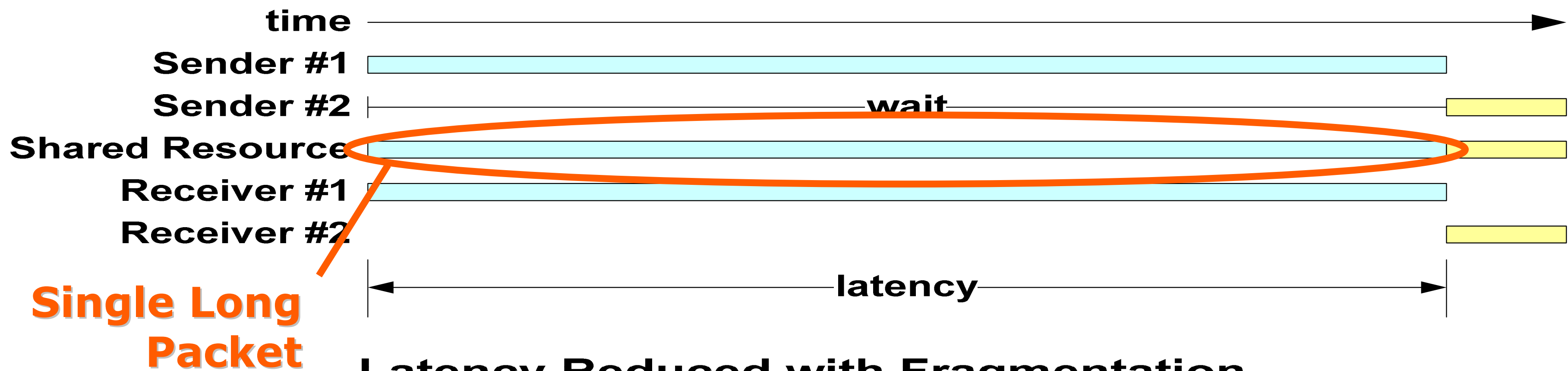


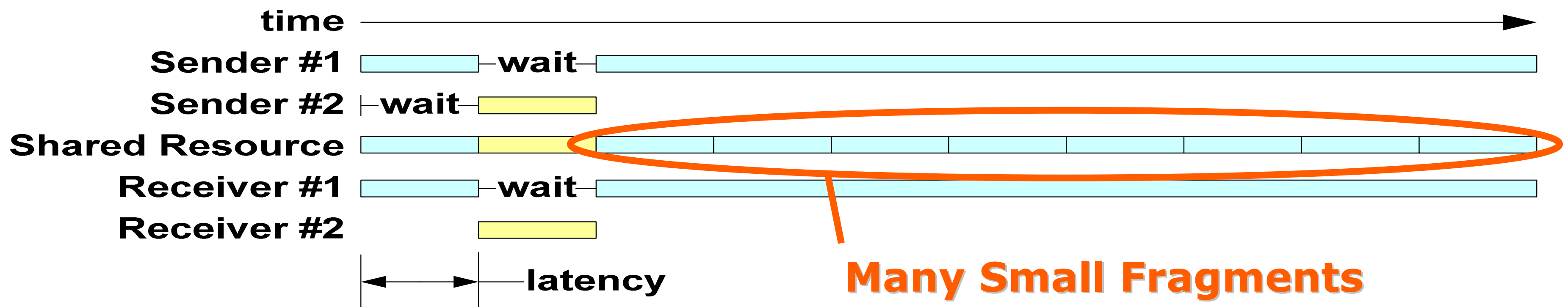
Full Matrix (shared bus) *Split Matrix (segmented bus)* *Sparse Matrix*

NoC Issues

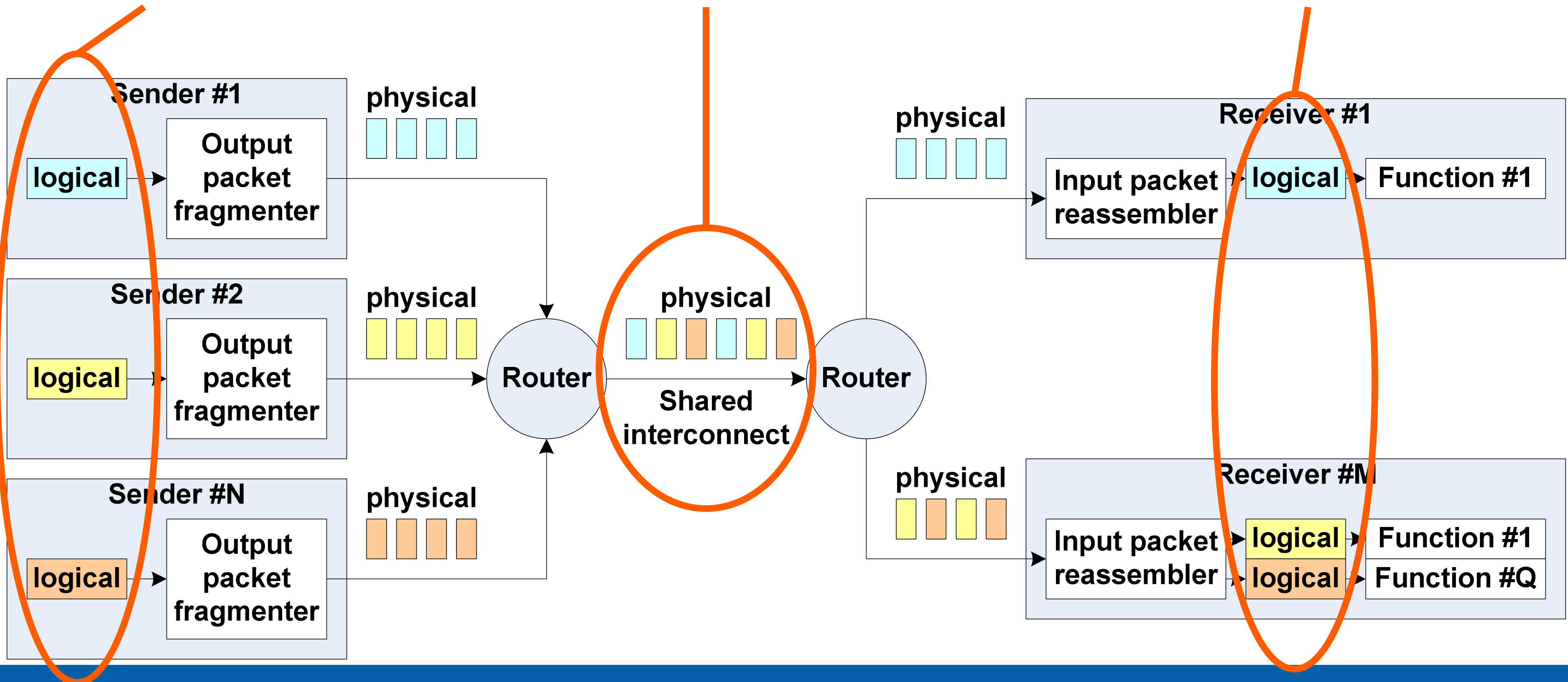
- Latency – caused by multiple streams contending for a shared interconnect
- Jitter – caused by time division multiplexing with variations in workload

Using Fragmentation to Constrain Latency

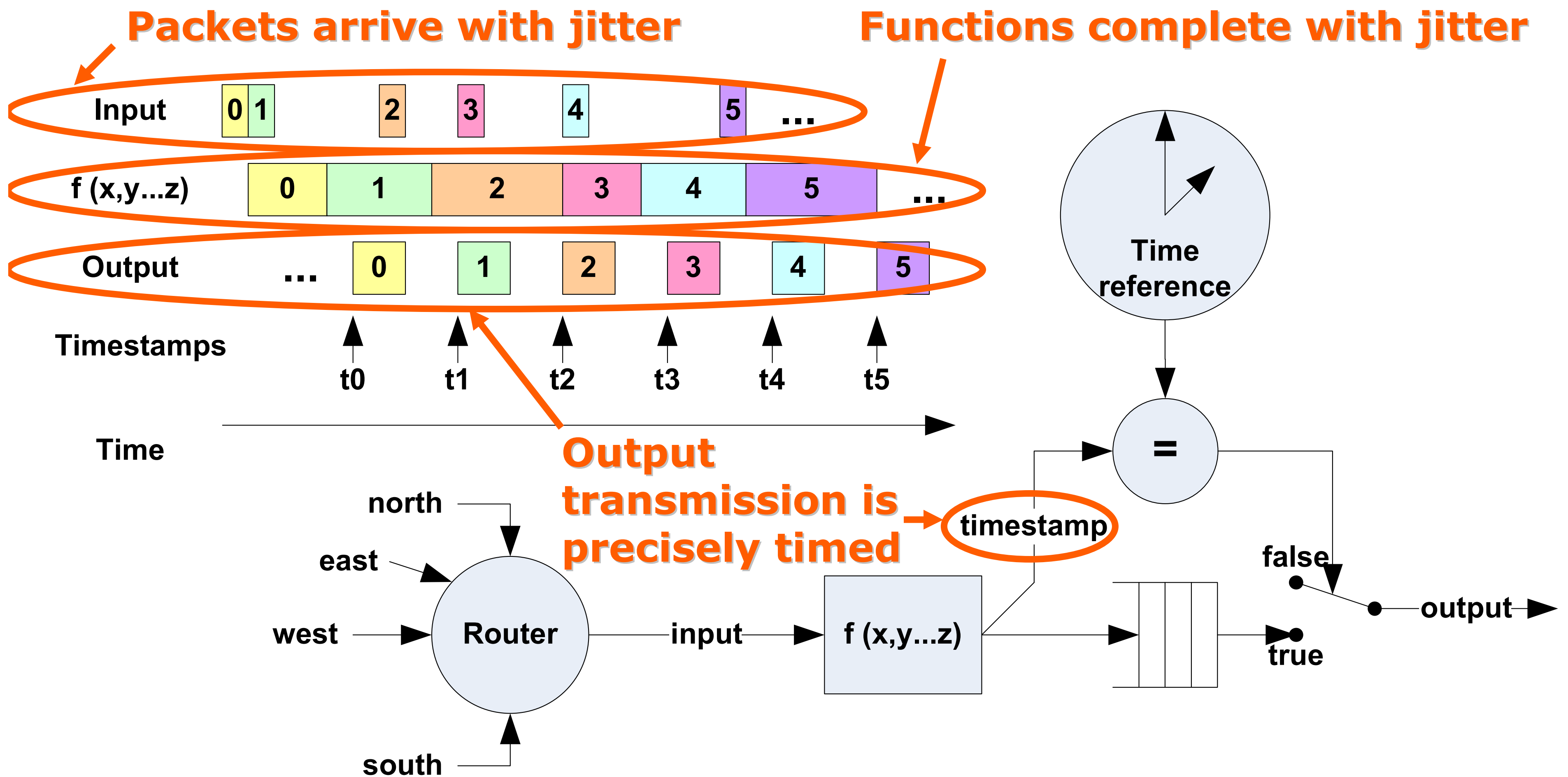




Using Time Division Multiplexing to Share Interconnect Segments

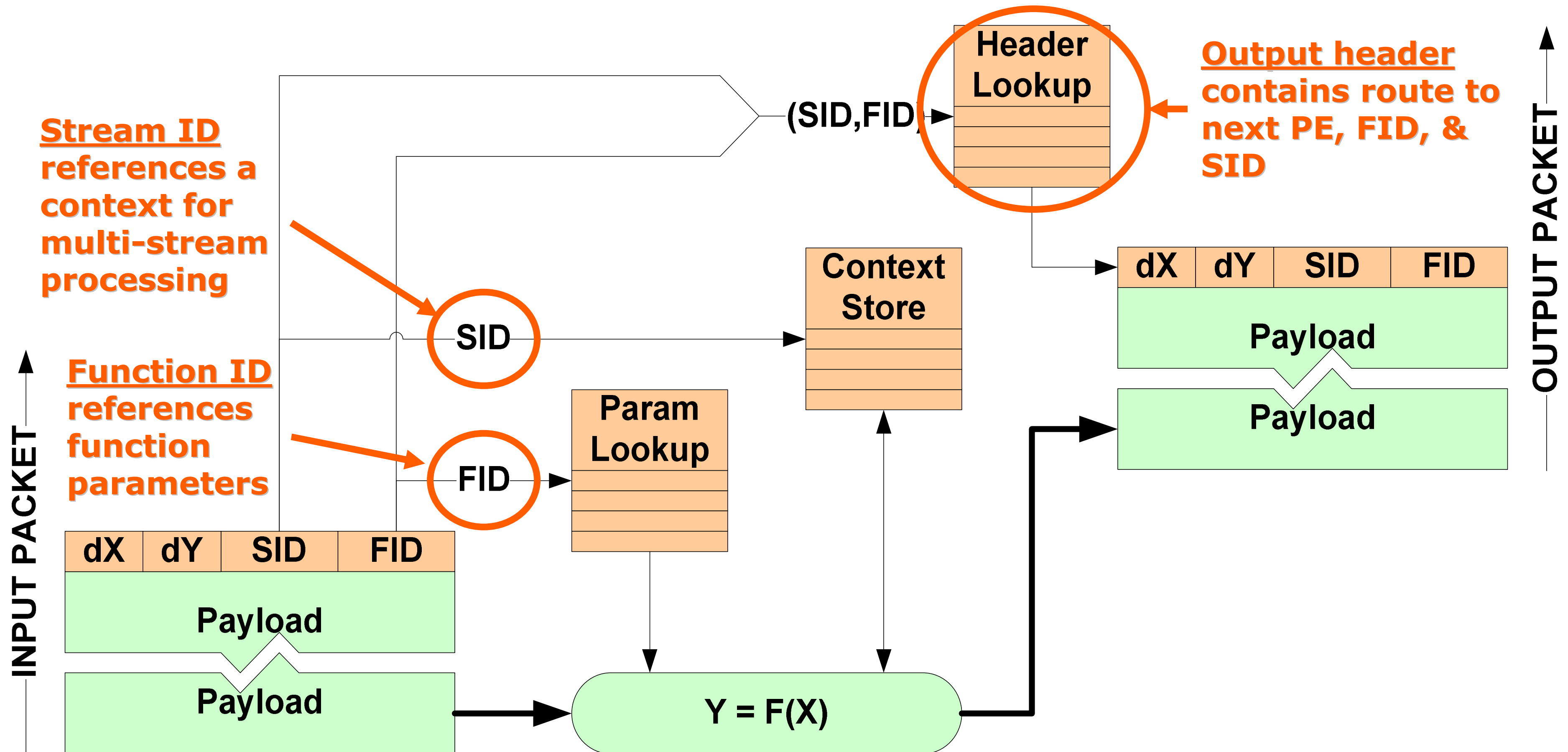


Using Timestamps to Constrain Jitter



intel

Data Driven Processing: Using a System of Tags to form Linked Lists



NoC Performance Requirements

Worst Case NoC Throughput:
(RX coded soft-bits @8 bits/soft-bit)

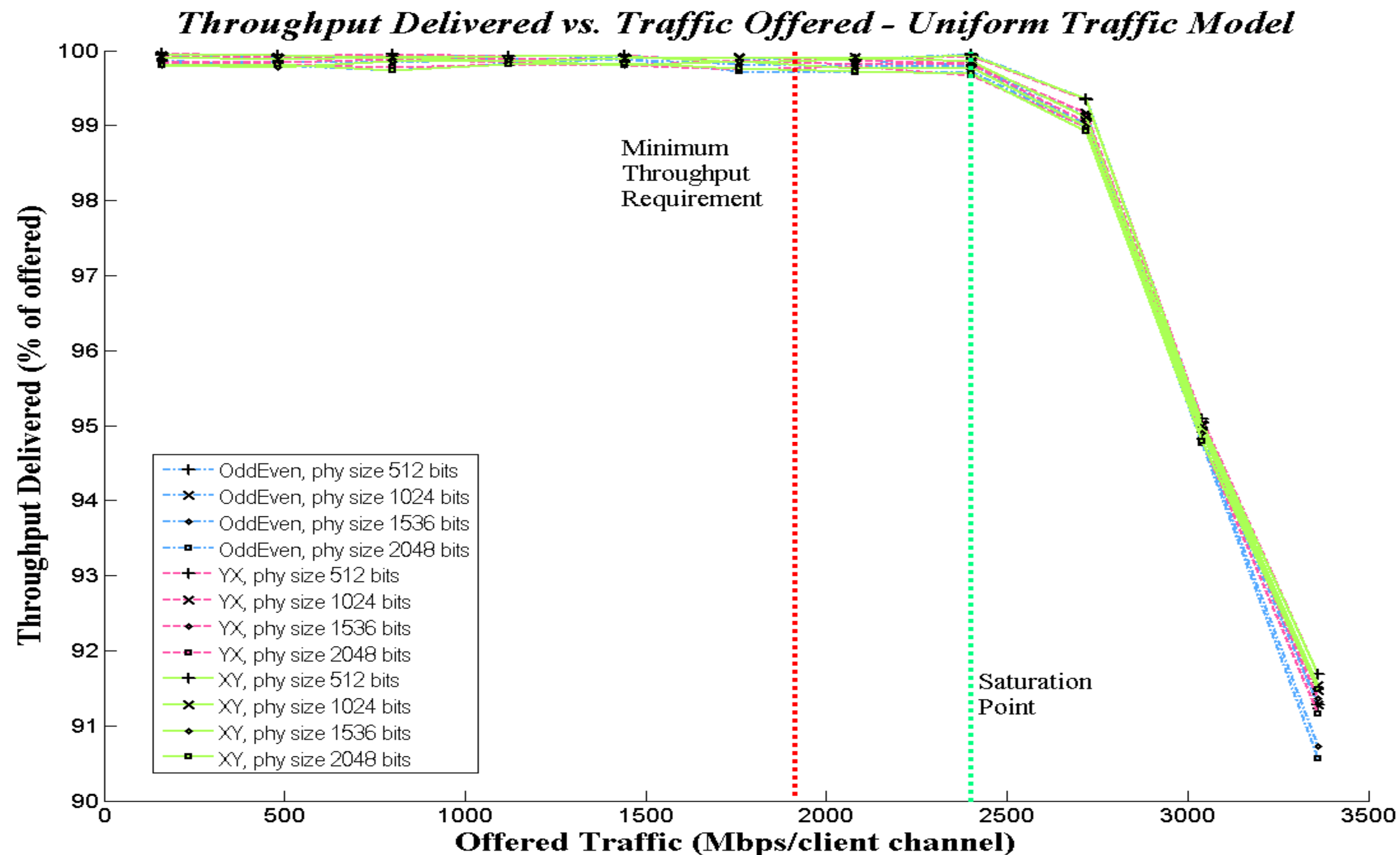
Protocol	**Throughput (Mbps)**
802.11n	1248
802.16e	336
DVB	314
per channel (aggregate)	**1898**

Worst Case NoC Latency:
(802.11n SIFS timing budget)

Budget	**Latency (μs)**
802.11n SIFS	16.0
MAC Budget PHY Budget	6.0 10.0
PE Budget NoC Budget	5.8 4.2
per channel (7 hops)	**0.6**

intel

Dimension Order Minimal Routing Satisfies Throughput Requirement



Latency is Constrained by Packet Size Not by Choice of Routing Algorithm

Latency vs. Offered Traffic - Routing Algorithm

OddEven, phy size 512 bits
OddEven, phy size 1024 bits
OddEven, phy size 1536 bits
OddEven, phy size 2048 bits
YX, phy size 512 bits
YX, phy size 1024 bits
YX, phy size 1536 bits
YX, phy size 2048 bits
XY, phy size 512 bits
XY, phy size 1024 bits
XY, phy size 1536 bits
XY, phy size 2048 bits

Saturation Point

Maximum Latency Limit

Latency (us)

2.0906
1.9266
1.7266
1.5266
1.3266
1.1266
0.9266
0.7266
0.5266
0.3266
0.1266

0 5 10 15 20 25 30 35 40 45

Offered Traffic (percentage of capacity %)

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

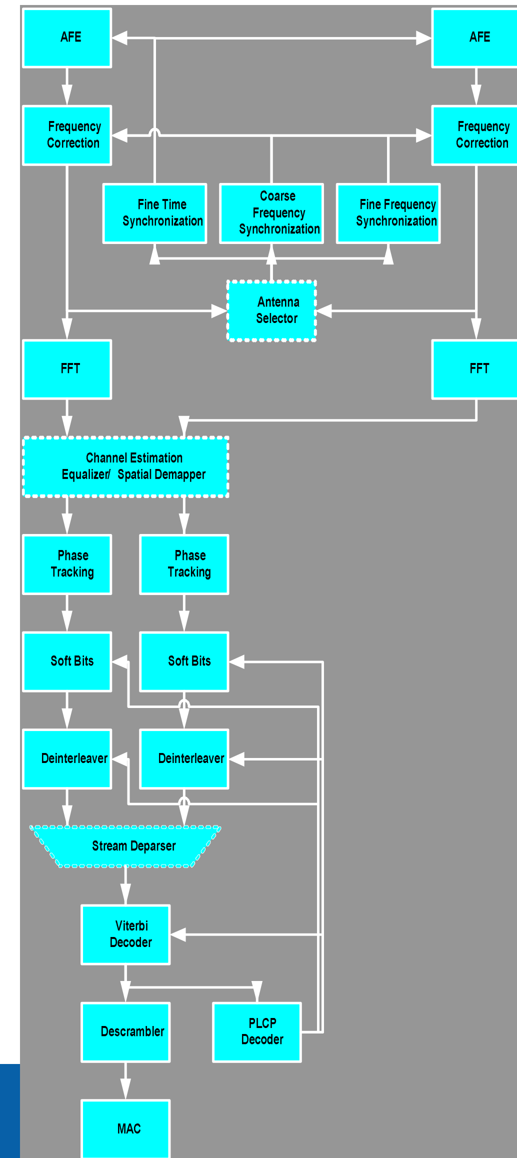
Programming Technology Challenges

- Vision: program the architecture as if it was a single DSP
  - We are not there yet
- Programming of heterogeneous accelerators
  - Degree of programmability varies i.e. DPE is more programmable than Viterbi decoder
  - Compilers for DPE and ILVPE
  - Other PEs are configured via registers
- Parallel programming model is in progress

Programming SCC

- <u>Map algorithms</u>
- Code algorithms
- Build
- Debug
- Simulate
- Profile

Mapping of 802.11n Rx

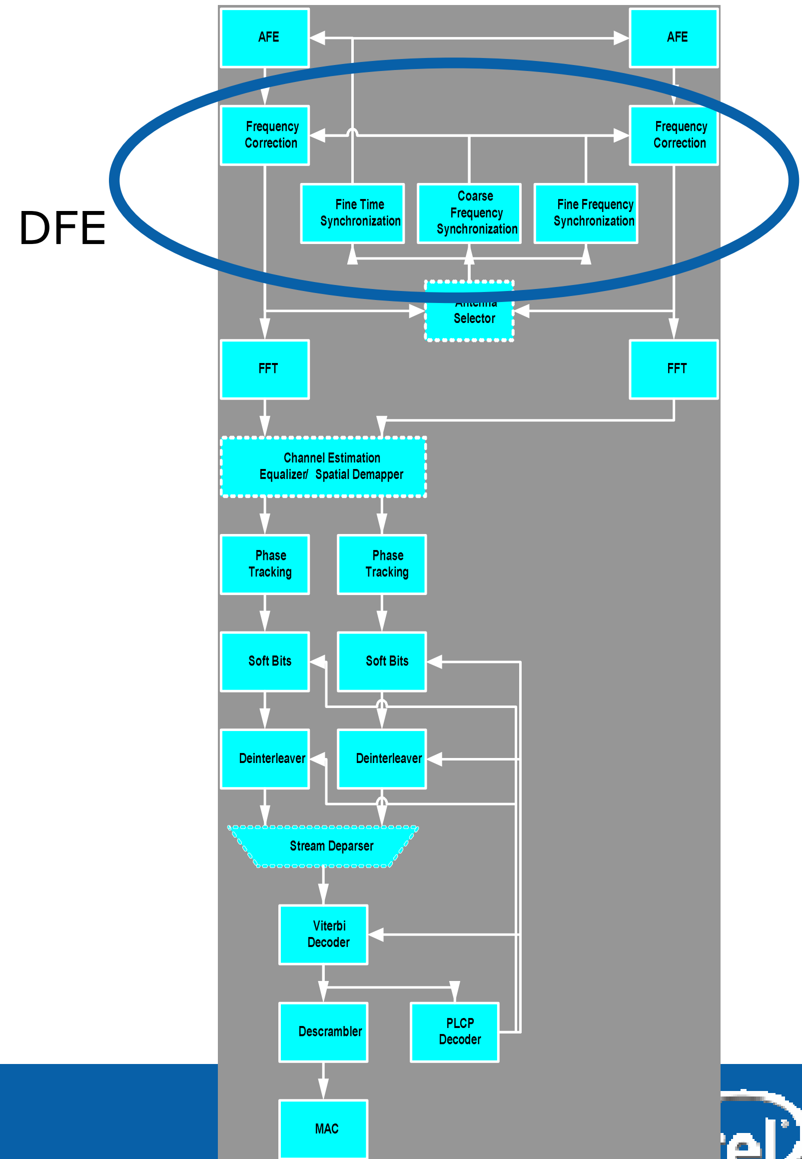


PPL (Parallel Programming Language) describes protocol mapping

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

DFE

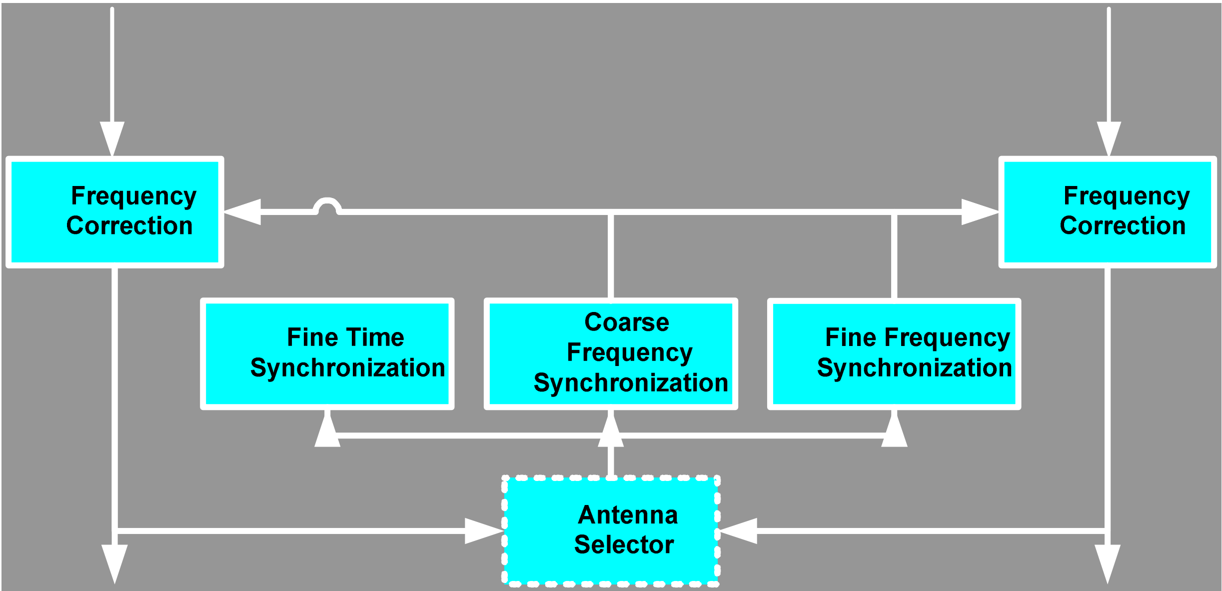


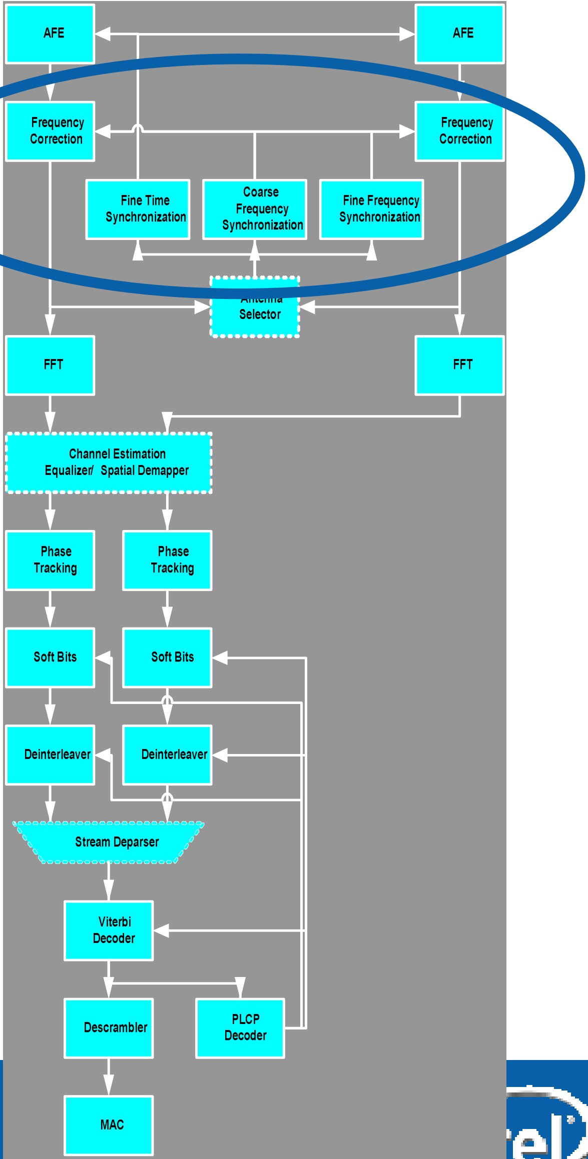
PPL (Parallel Programming Language) describes protocol mapping

Programming SCC

- Map algorithms

DFE



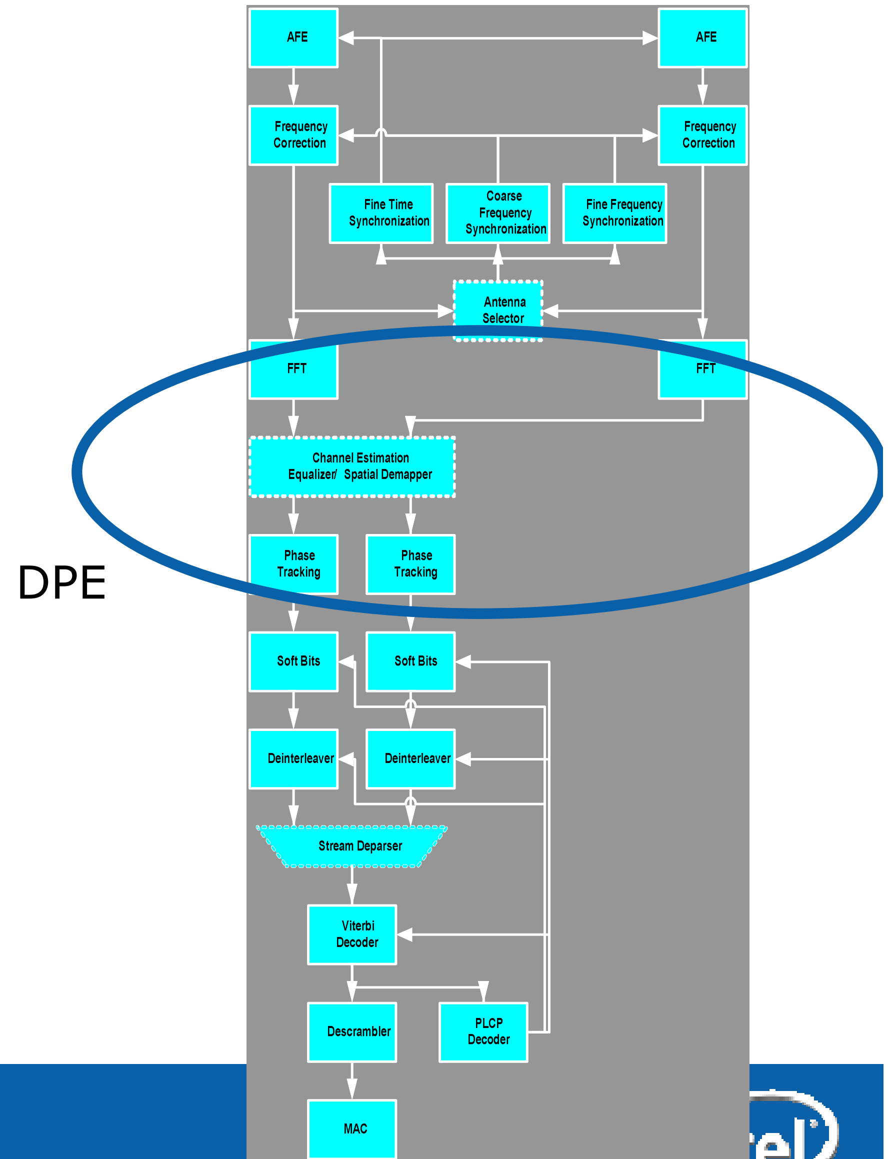


Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

PPL (Parallel Programming Language) describes protocol mapping

DPE



Programming SCC

- Map algorithms

PE

FFT

FFT

Channel Estimation
Equalizer/ Spatial Demapper

Phase Tracking

Phase Tracking

Soft Bits

Soft Bits

AFE

AFE

Frequency Correction

Frequency Correction

Fine Time Synchronization

Coarse Frequency Synchronization

Fine Frequency Synchronization

Antenna Selector

FFT

FFT

Channel Estimation
Equalizer/ Spatial Demapper

Phase Tracking

Phase Tracking

Soft Bits

Soft Bits

Deinterleaver

Deinterleaver

Stream Deparser

Viterbi Decoder

Descrambler

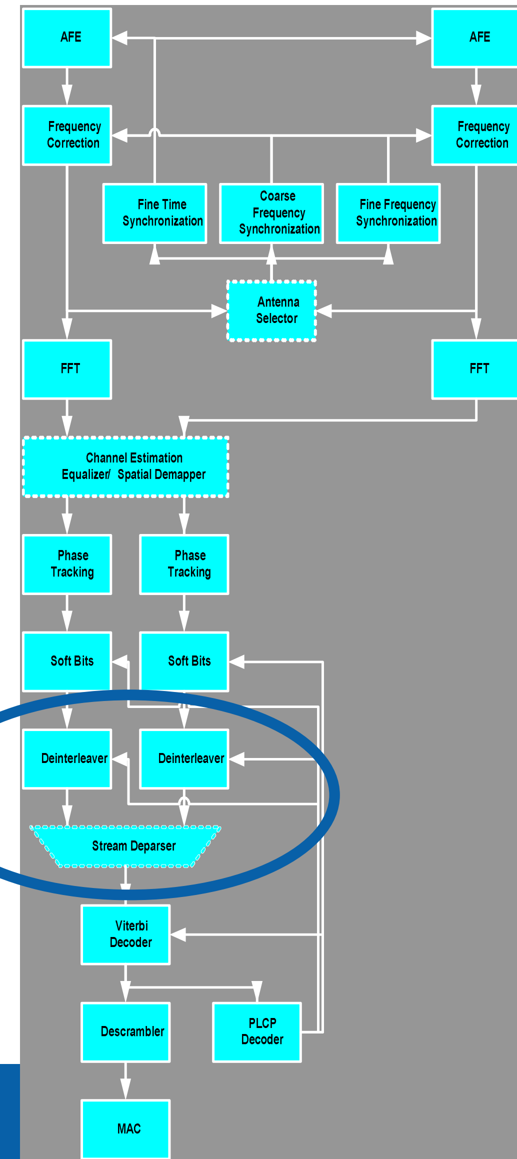
PLCP Decoder

MAC

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

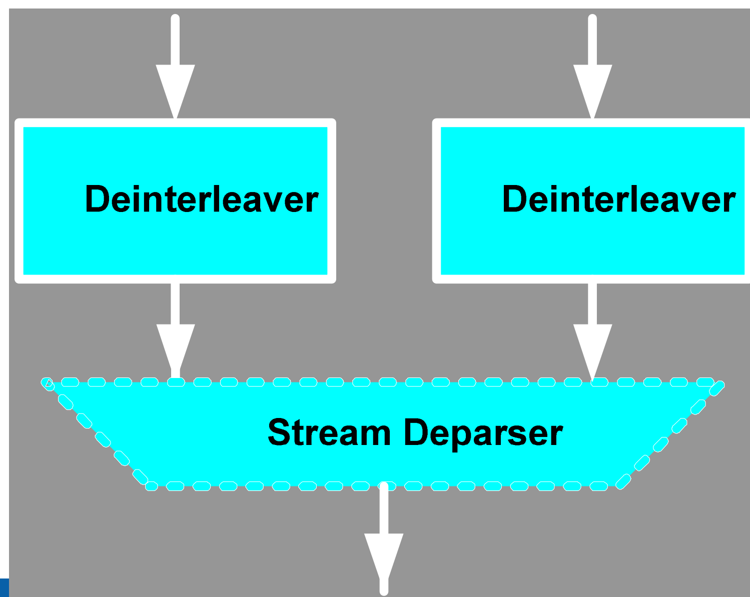
ILVPE

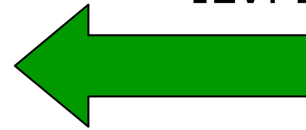


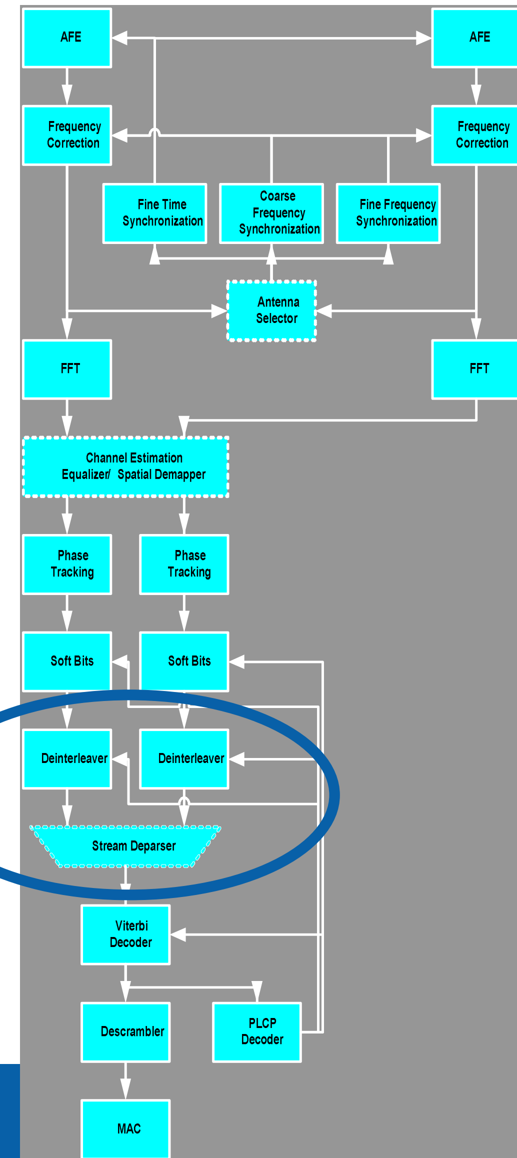
PPL (Parallel Programming Language) describes protocol mapping

Programming SCC

- Map algorithms





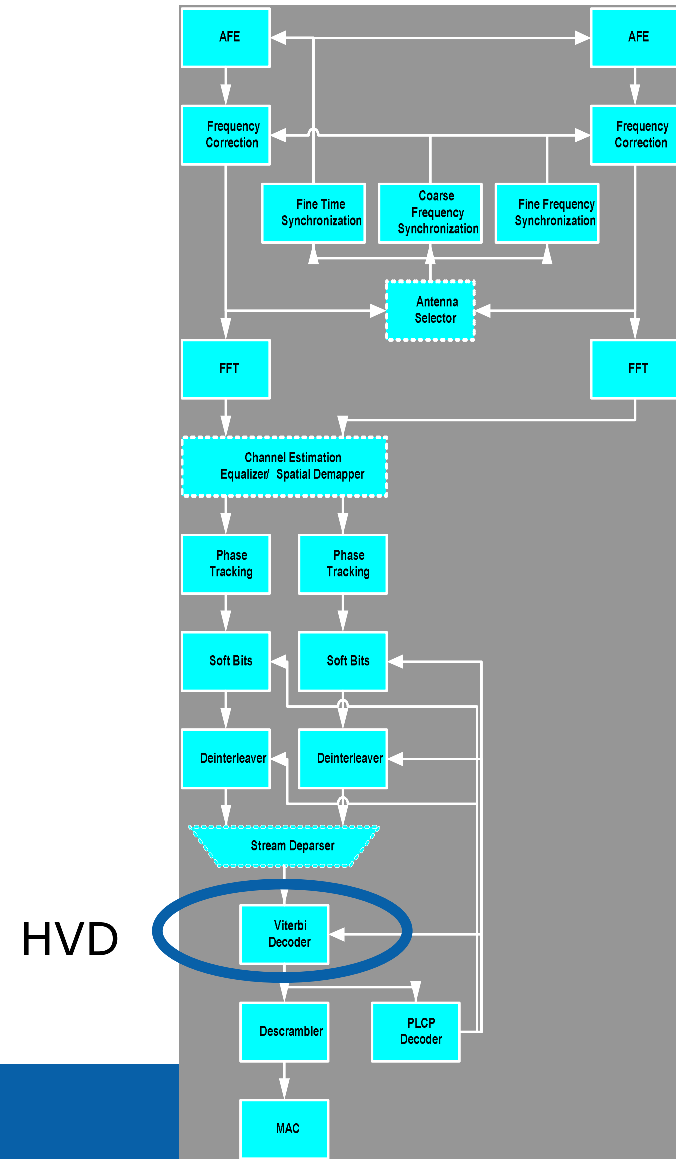


Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

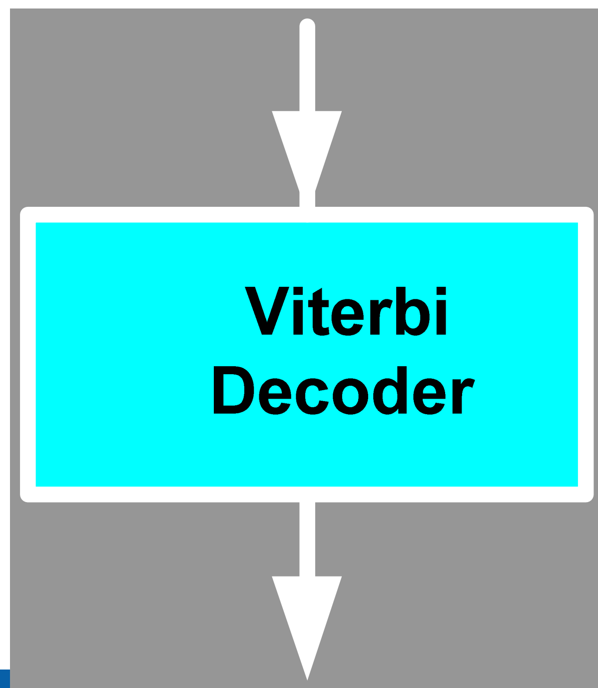
PPL (Parallel Programming Language) describes protocol mapping

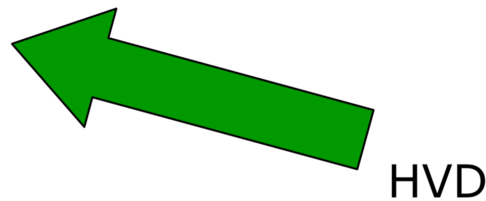
HVD



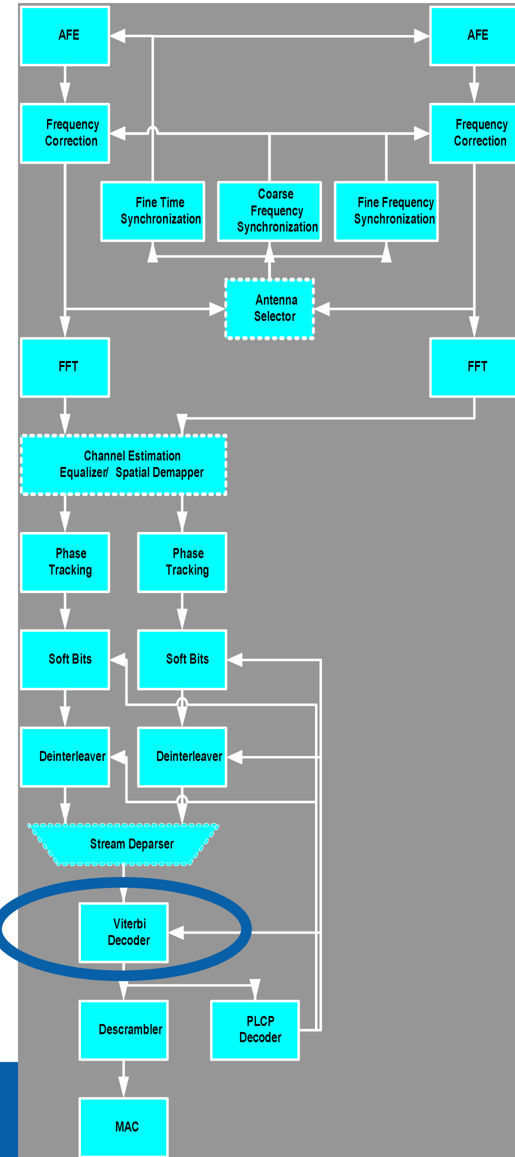
Programming SCC

- Map algorithms





HVD

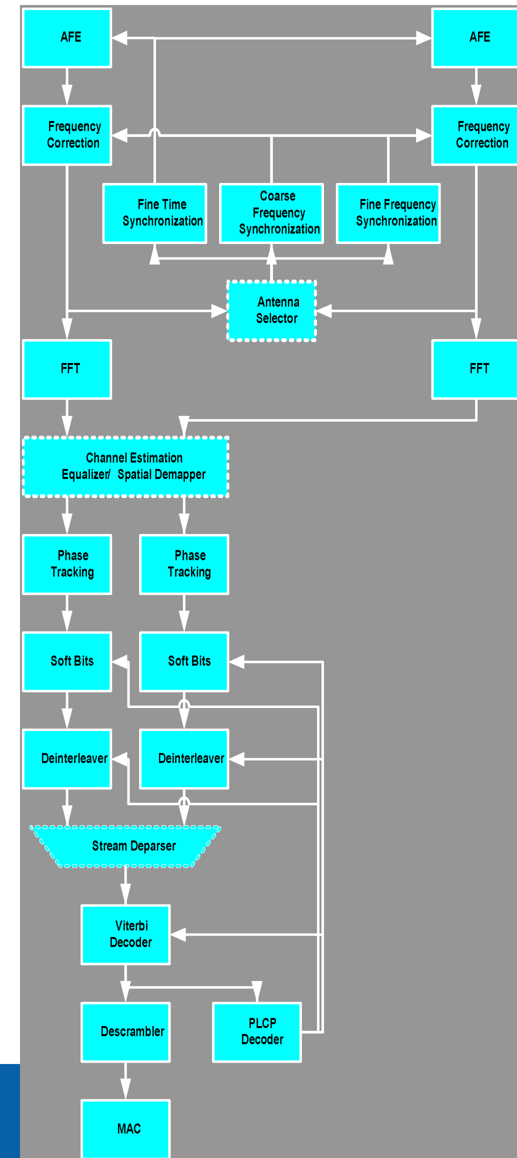


Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

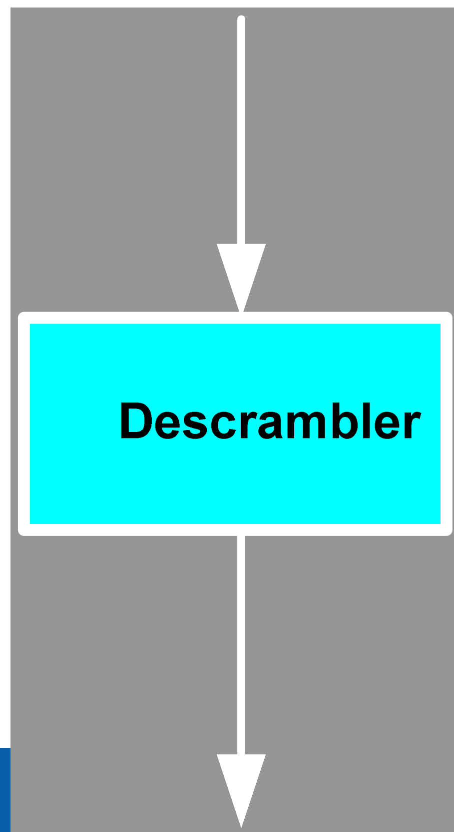
PPL (Parallel Programming Language) describes protocol mapping

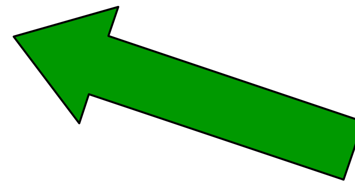
CCPE



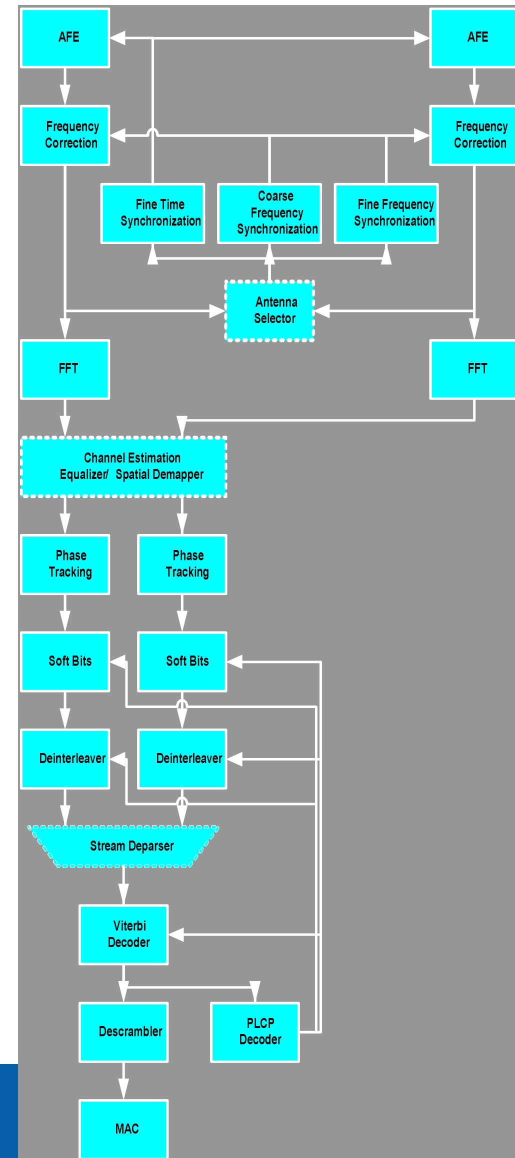
Programming SCC

- Map algorithms





CCPE

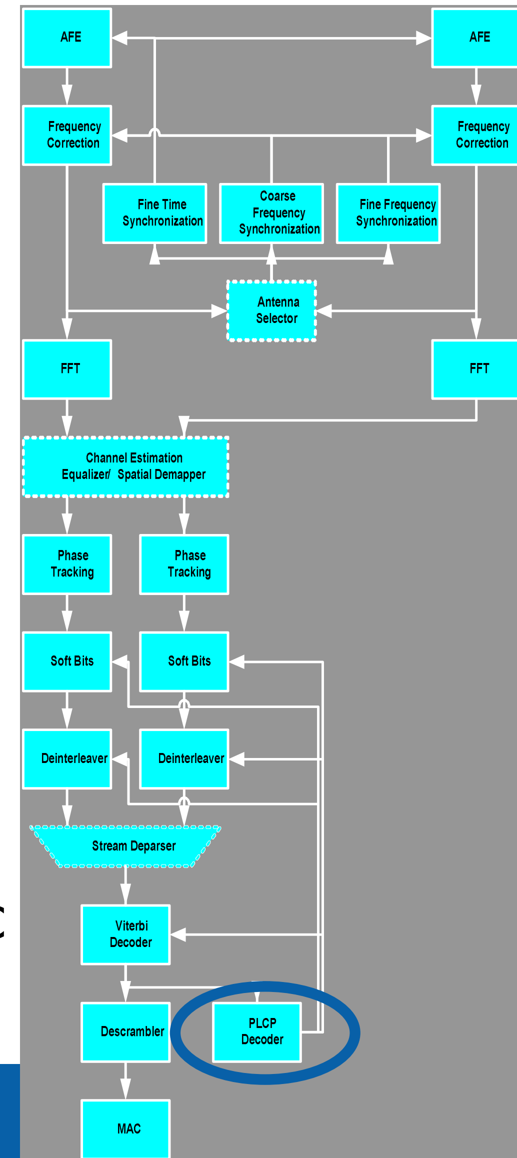


Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

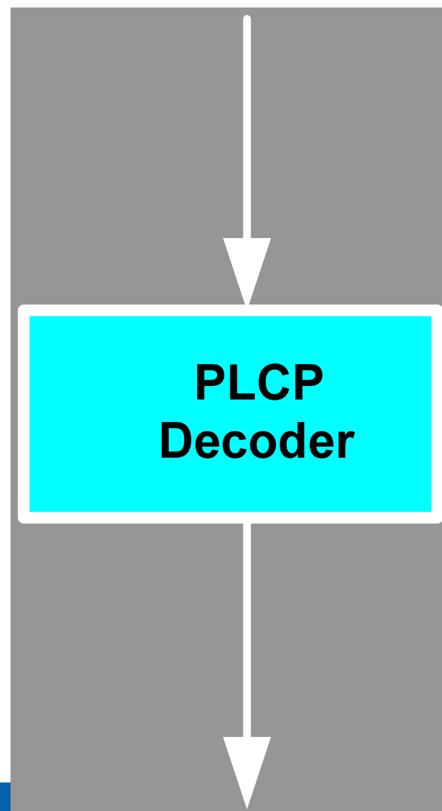
PPL (Parallel Programming Language) describes protocol mapping

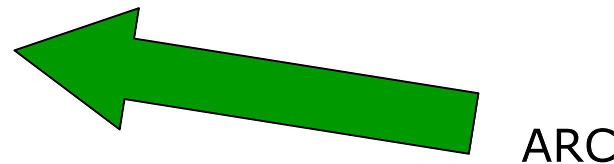
ARC



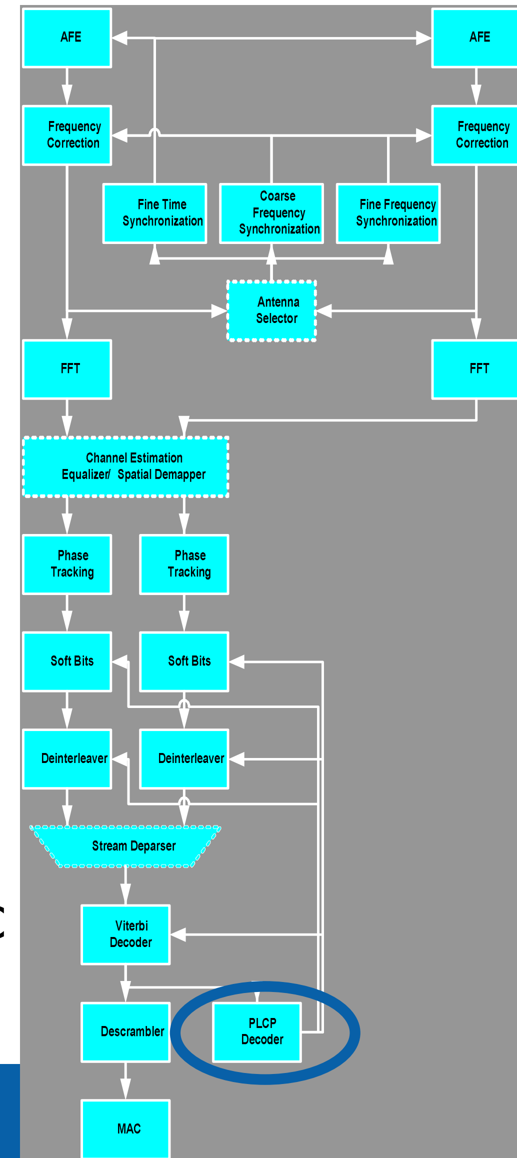
Programming SCC

- Map algorithms





ARC

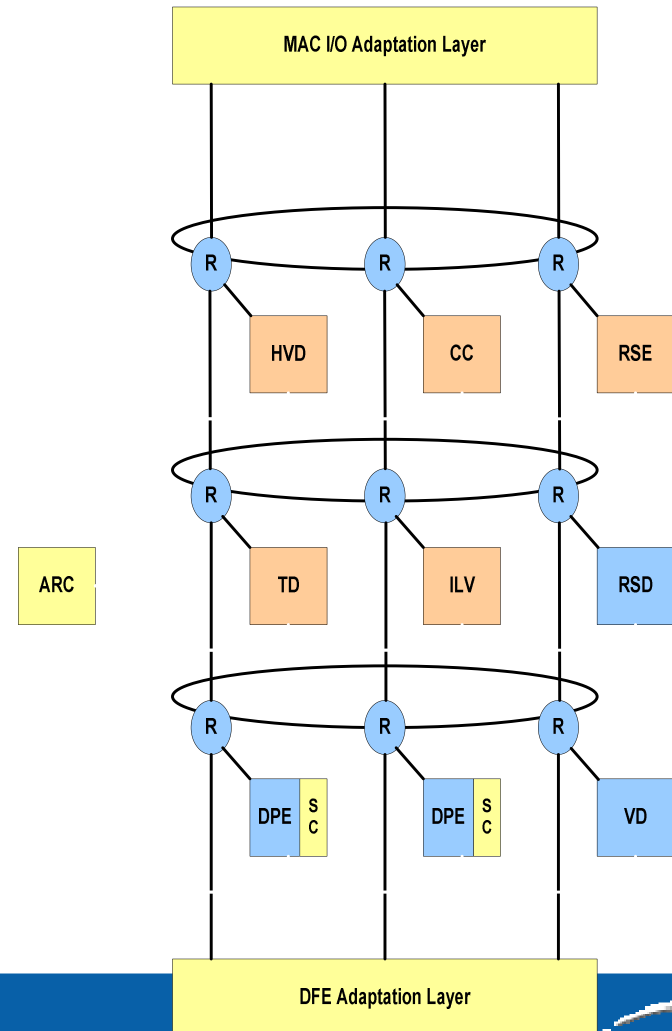


Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

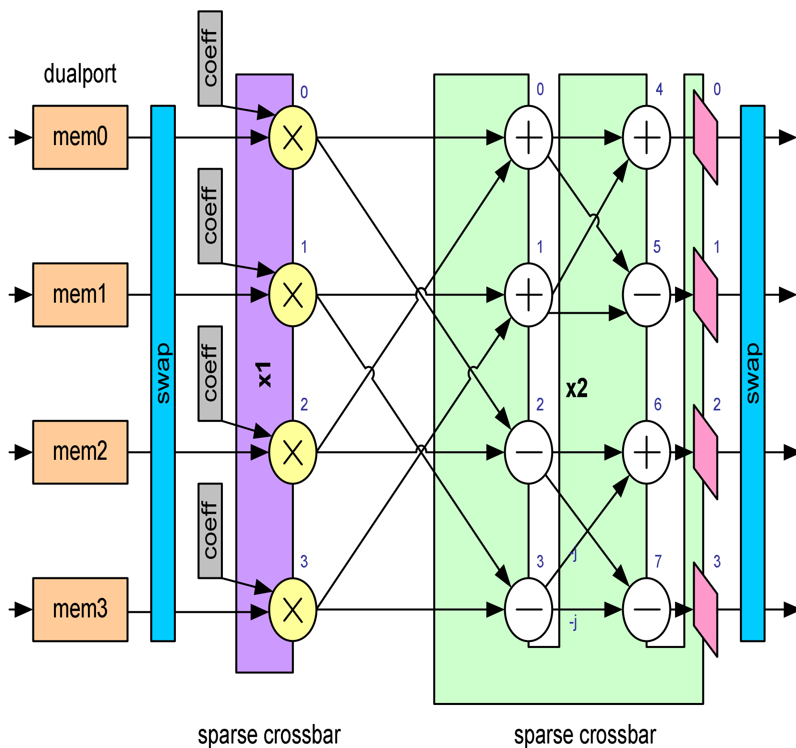
PPL (Parallel Programming Language) describes protocol mapping

Algorithms mapped to PEs

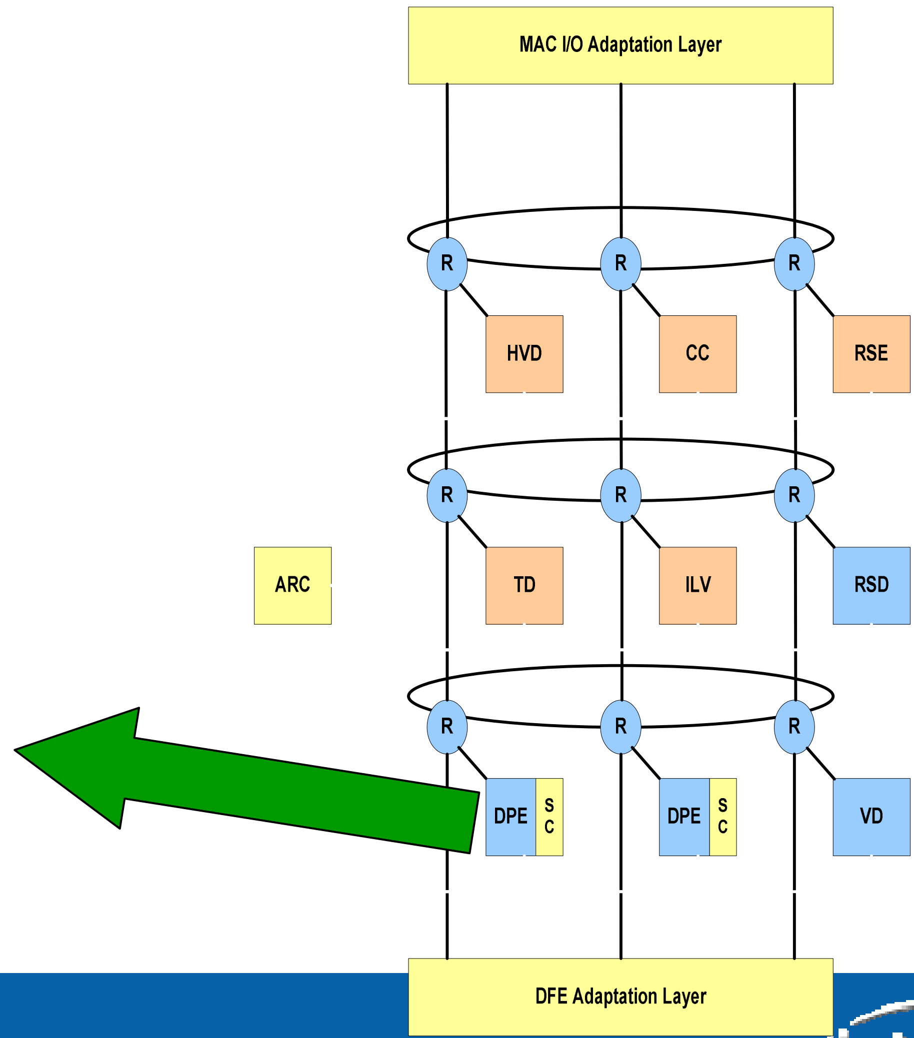


Programming SCC

- Map algorithms



Algorithms mapped to PEs

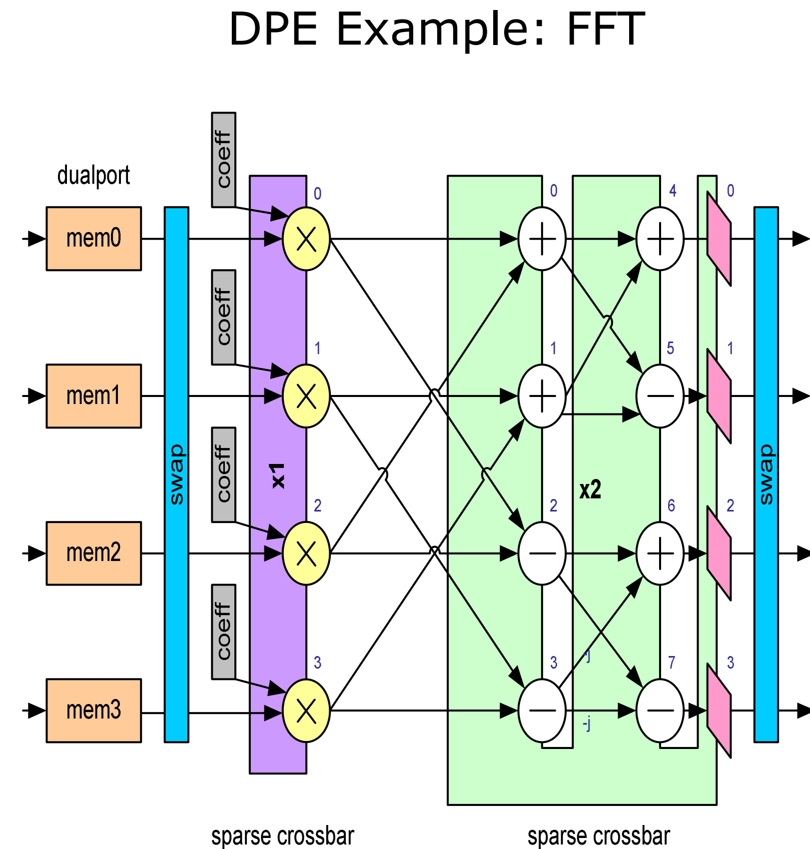


intel

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

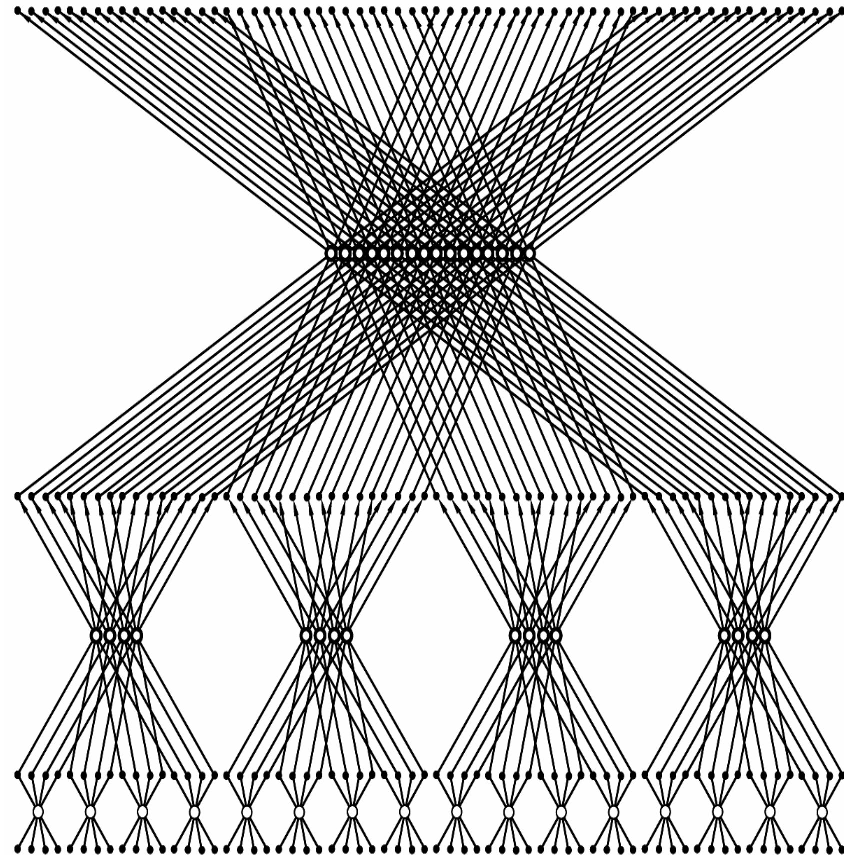
DPE Example: FFT



DPE configuration for 64 pt radix-4 FFT

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile



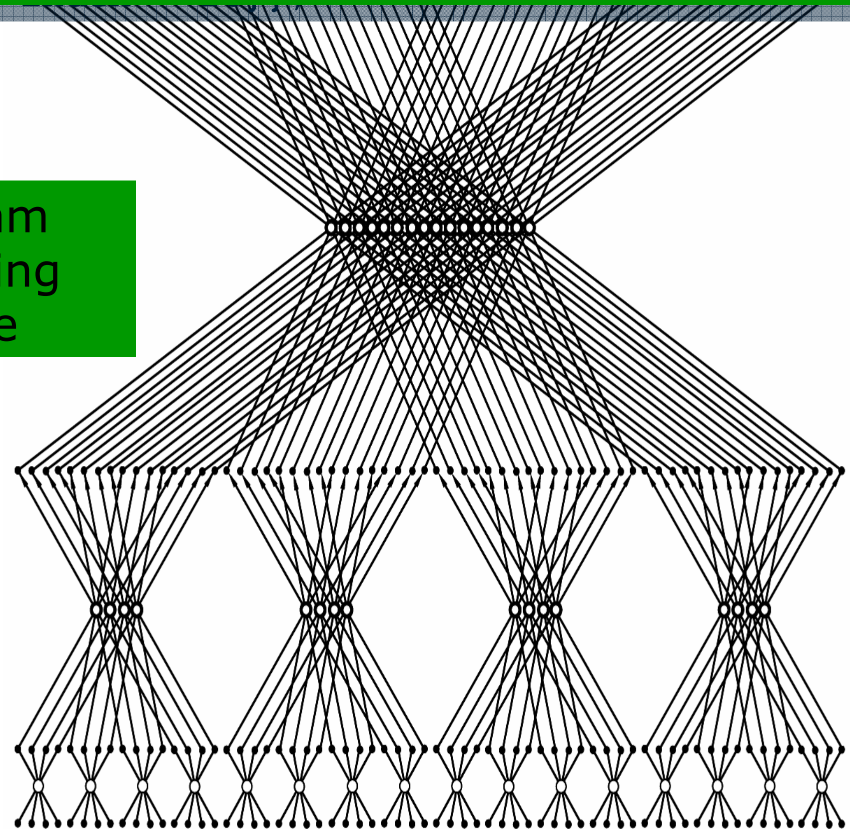
DPE Example: FFT

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

```
selector[iter:16] dataSelector1[index:4] = {{
iter + index * 16}};
```

Data stream Programming Language



DPE Example: FFT

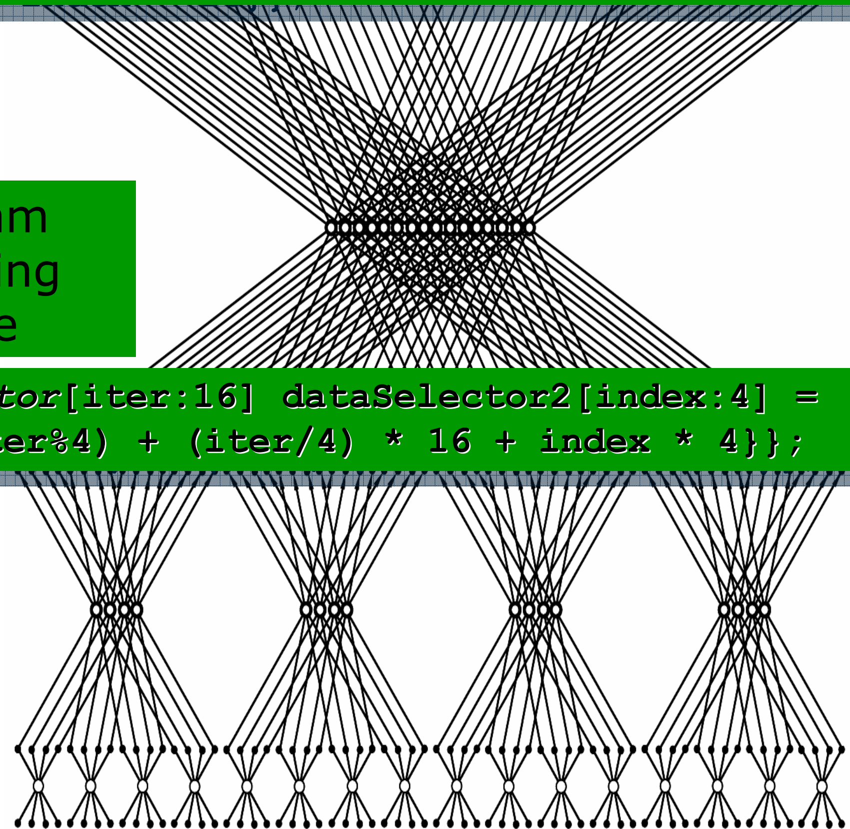
Programming SCC

- Map algorithms
- <u>Code algorithms</u>
- Build
- Debug
- Simulate
- Profile

```
selector[iter:16] dataSelector1[index:4] = {{
iter + index * 16}};
```

Data stream Programming Language

```
selector[iter:16] dataSelector2[index:4] =
{{(iter%4) + (iter/4) * 16 + index * 4}};
```

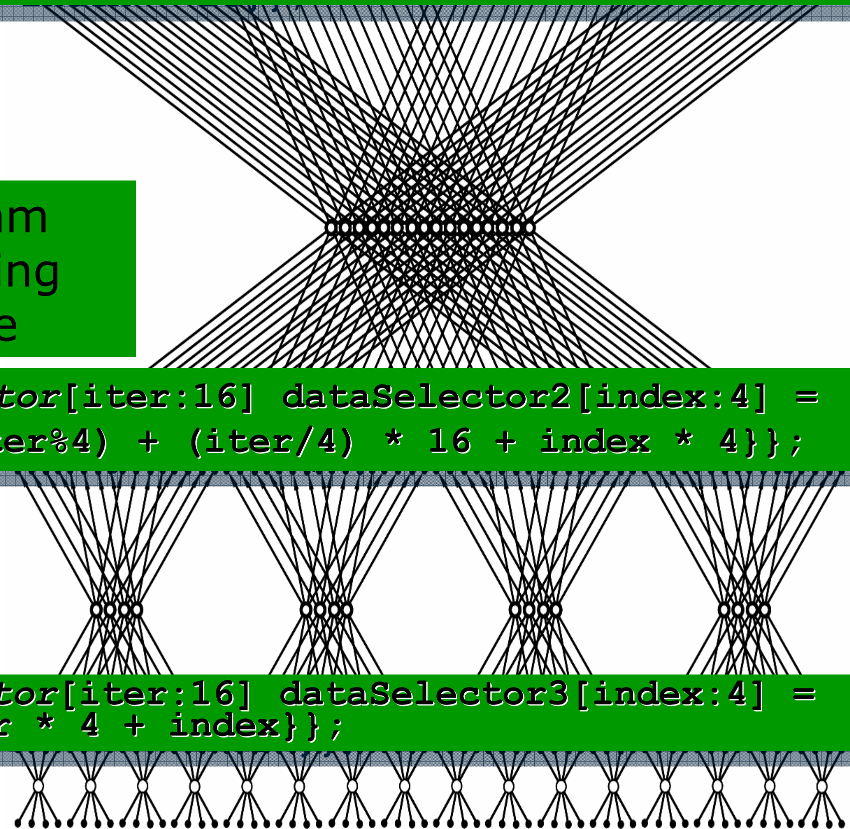


DPE Example: FFT

Programming SCC

- Map algorithms
- <u>Code algorithms</u>
- Build
- Debug
- Simulate
- Profile

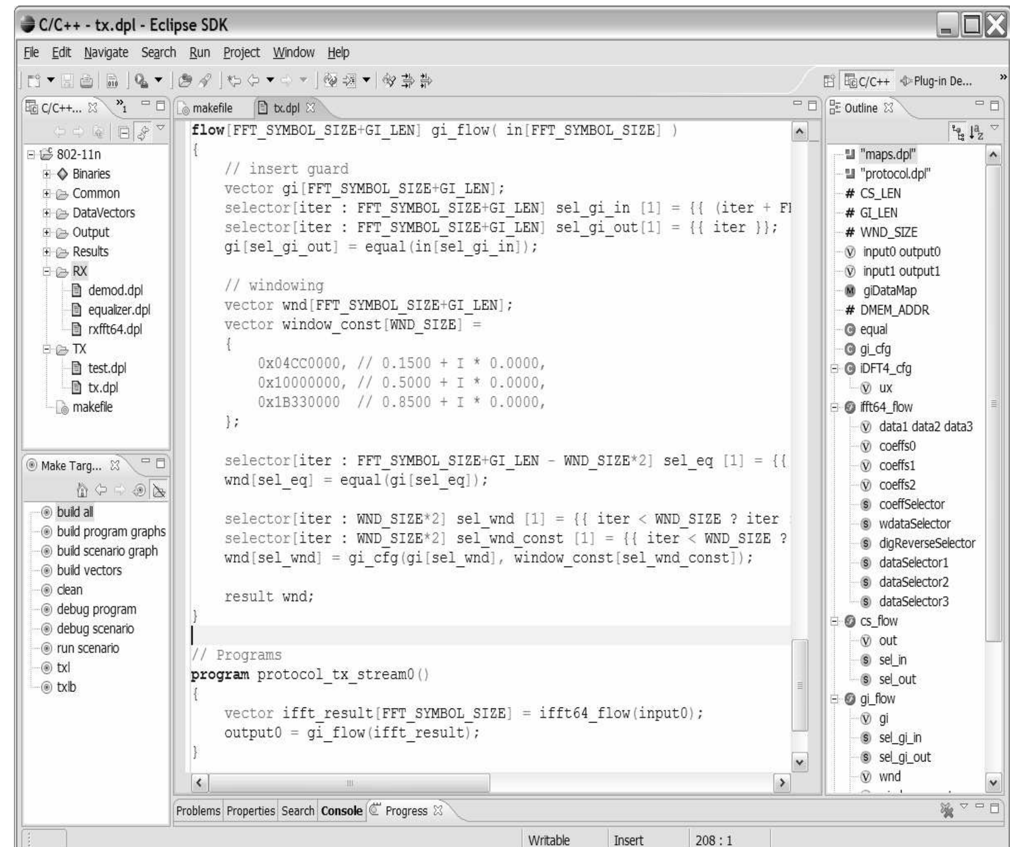
Data stream Programming Language



DPE Example: FFT

Programming SCC

- Map algorithms
- <u>Code algorithms</u>
- Build
- Debug
- Simulate
- Profile



intel

Programming SCC

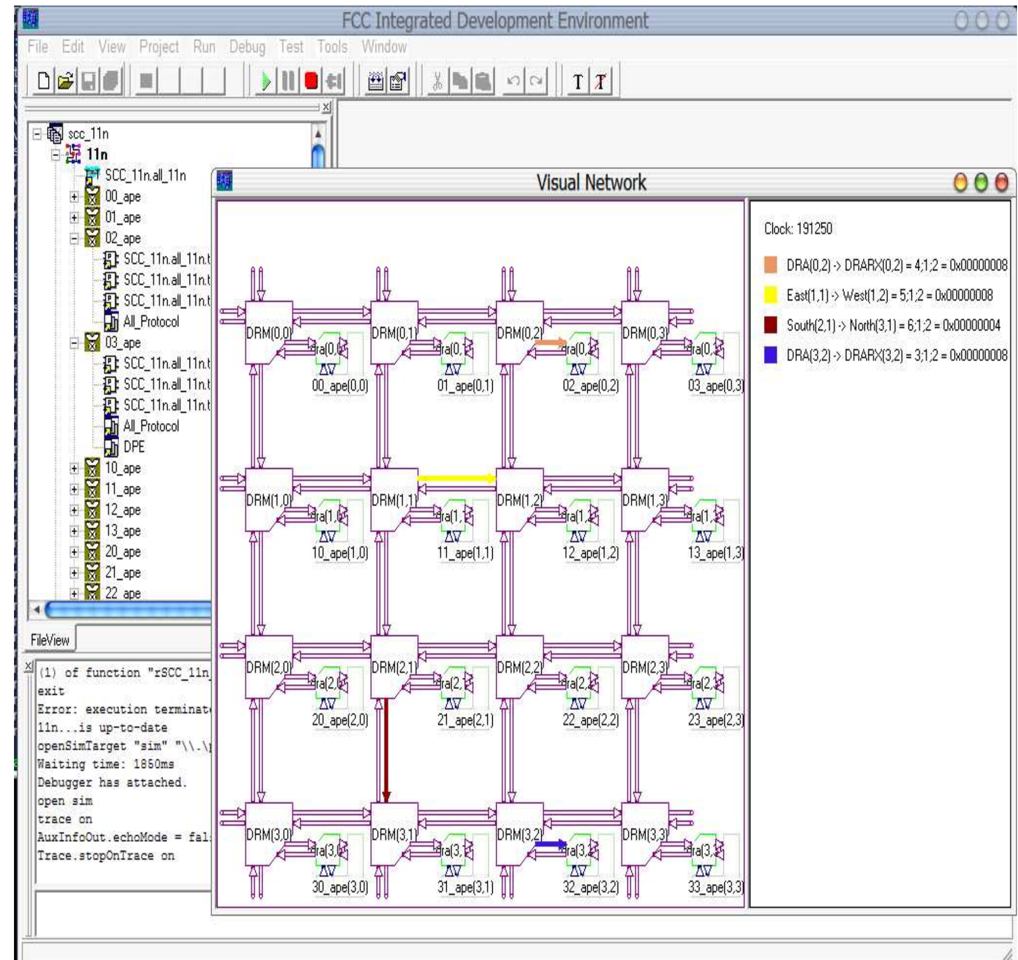
- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- Simulate
- Profile

Programming SCC

- Map algorithms
- Code algorithms
- Build
- Debug
- <u>Simulate</u>
- Profile

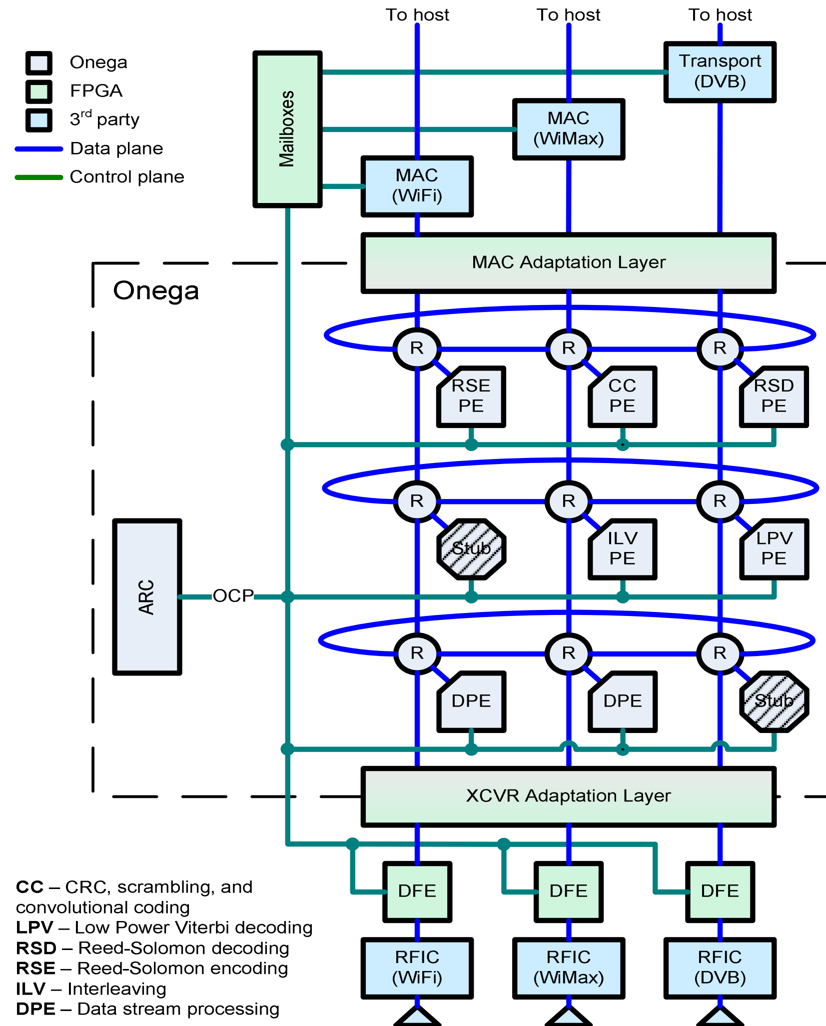


Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

Onega Test Chip

- 65nm process
- Taped out in Dec 2007
- Subset of PEs included

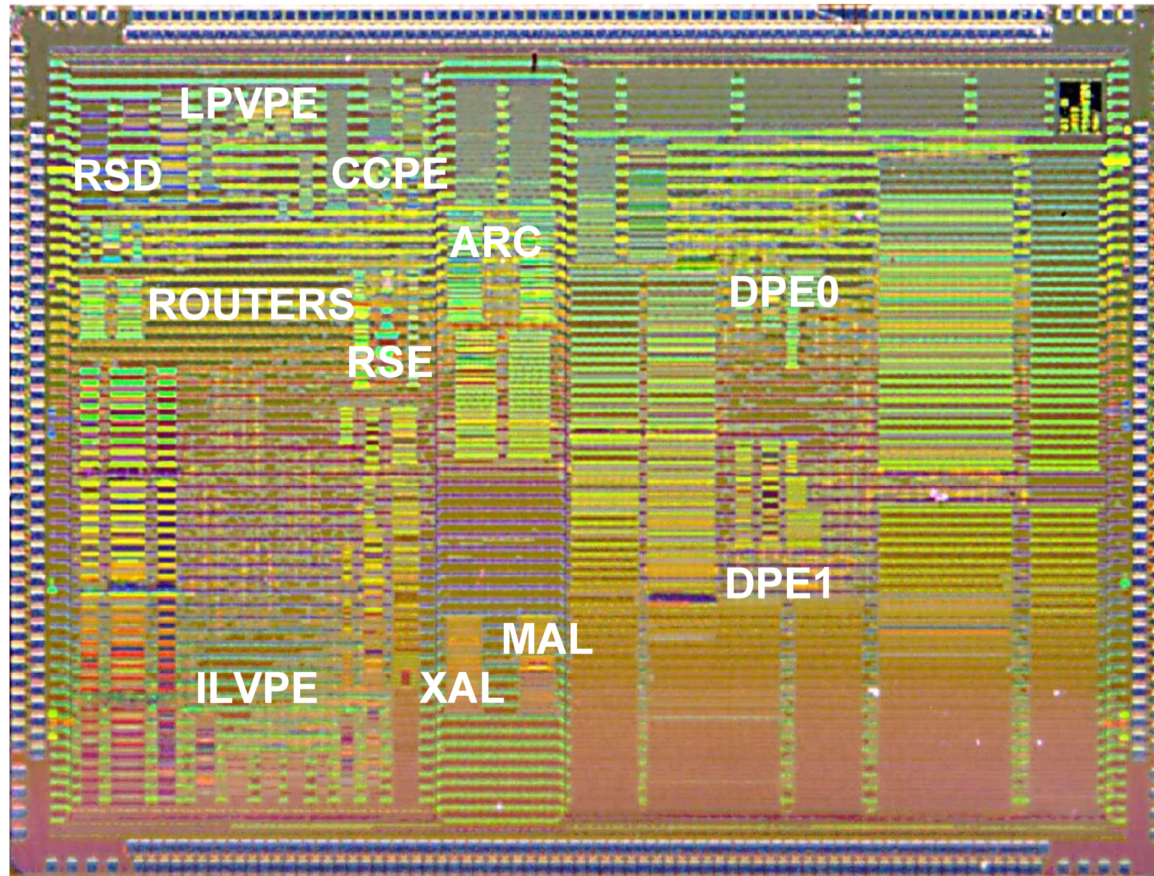


intel

Onega Die Photo

5.77 mm

4.26 mm



intel

Silicon Results

- Process technology: 65nm
- Silicon area (excl. pads): 20.75mm^2
- Program memory-DPE1, DPE2 and microcontroller: 96+96+32=224kbytes
- Data memory-DPE1, DPE2 and microcontroller: 16+256+8=280kbytes
- Logic gate count: 1.36M
- Supply Voltage: 1.1V Core, 3.3V I/O
- Package: WB-PBGA 31x31 mm
- Signal I/O Count: 332
- Measured Clock frequency: 233MHz
- Paper summarizing power measurements has been submitted to *ISVLSI* 2009

Areas of Processing Elements

Label	Classification	Area mm^2
DFE	AGC, resample, filter, detect	3.600
DPE1	64-point FFT/IFFT, chn eq, QAM	2.13
DPE2	8k-point FFT/IFFT, chn eq, QAM	4.69
ILV	puncture, interleave, multiplex	1.57
LPV	Low power Viterbi decoding	0.21
RSD	RS decoding	0.26
RSE	RS encoding	0.09
CCPE	CRC, randomization, coding	0.21
NoC	Interconnect	0.091
ARC	Configuration and control	0.73

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

Protocol Implementations (to date)

- 802.11a/n
  - subset of MCSs (limited by Viterbi decoder rate of 54 Mbps) tested on Onega silicon
  - 16 μs SIFS requirement met
- 802.16e: range of modes validated on Onega silicon
- DVB-H: Rx on RTL simulator
- Bluetooth: modulation and demodulation on DPE simulator
- GPS: code acquisition and tracking on DPE simulator

802.11a High Rate Inter-Symbol Control

RFIC DFE DPE ILV LPV CC ARC MAC

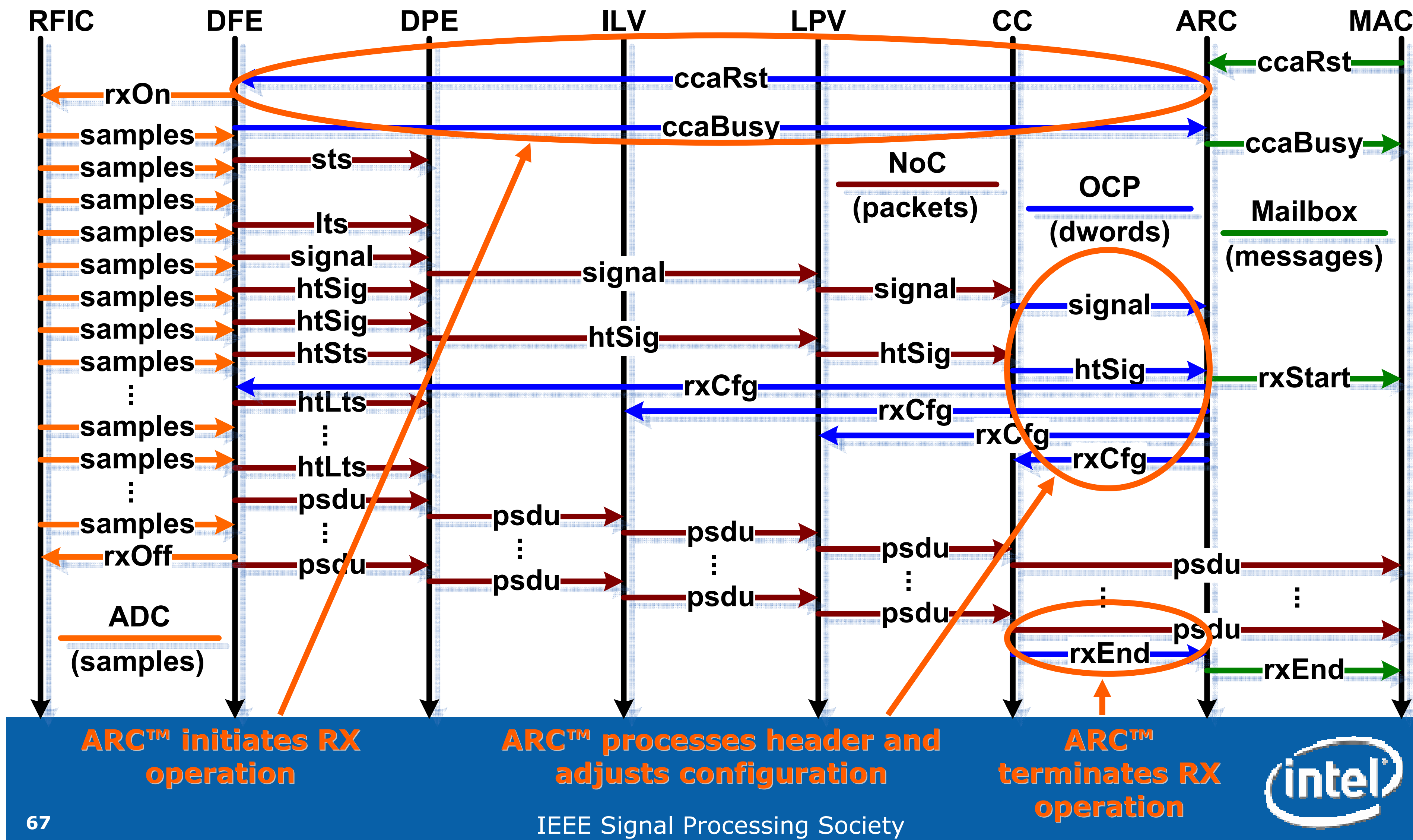
ccaRst
ccaRst
rxOn
samples
ccaBusy
ccaBusy
sts
lts
signal
htSig
htSig
htSts
rxCfg
htLts
htLts
psdu
psdu
rxOff
NoC (packets)
OCP (dwords)
Mailbox (messages)
signal
signal
signal
htSig
htSig
htSig
rxStart
rxCfg
rxCfg
rxCfg
psdu
rxEnd
rxEnd
ADC (samples)

Symbols arrive at 250kHz rate tagged by type

Header delivered to ARC™

Payload delivered to MAC

802.11a Low Rate Inter-Frame Control



ARC™ initiates RX operation

ARC™ processes header and adjusts configuration

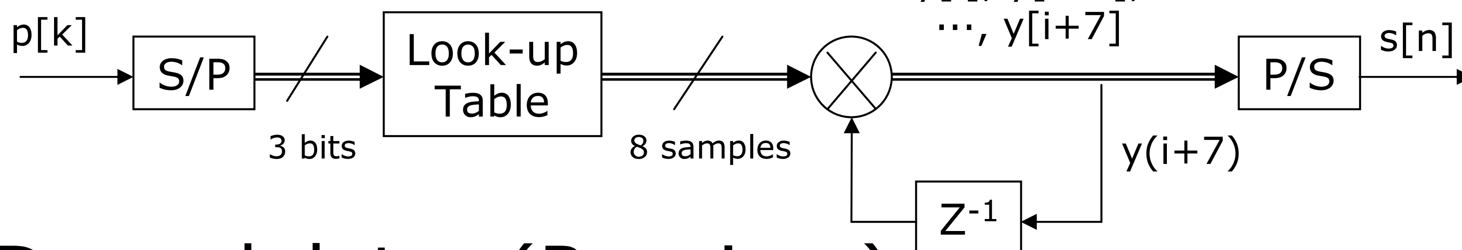
ARC™ terminates RX operation

intel

GFSK modulator / demodulator

Modulator (Transmitter)

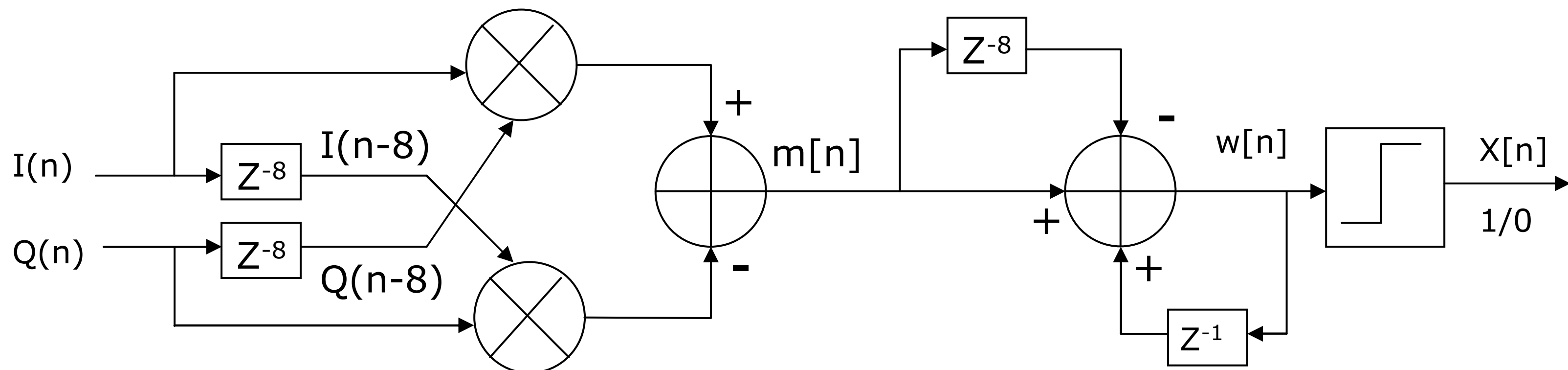
Choose Tx sampling rate as 8 Ms/s.



Demodulator (Receiver)

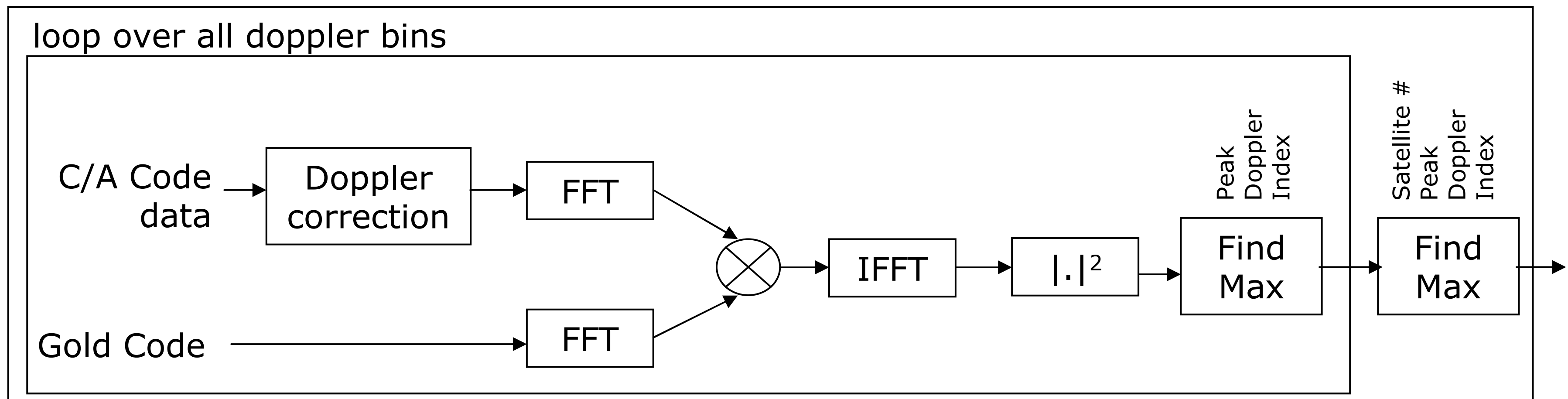
Choose Rx sampling rat

Demodulator operates at 200 kbps (goal is 1 Mbps)

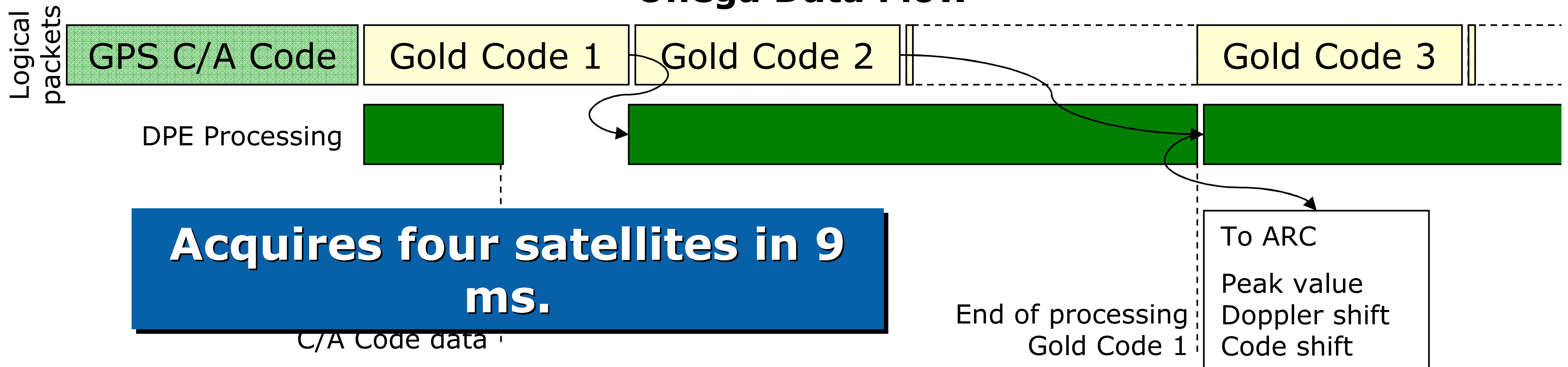


intel

GPS Code Acquisition



Onega Data Flow



Acquires four satellites in 9 ms.

intel

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- <u>Learnings</u>
- Summary

Learnings

- Heterogeneous coarse-grained PE NoC architecture validated as real-time wireless baseband
- Area and power competitive with fixed solutions
- Stream programming model and tools developed
- General parallel programming tool for entire set of PEs remains a goal

Agenda

- Introduction
- Motivation
- Architecture
- Programming
- Test Chip
- Implementation Examples
- Learnings
- Summary

Summary

- We have demonstrated a flexible radio baseband
- Taped out test chip, programmed it, validated and measured power
- Next steps
  - Implement additional protocols
  - Improvements to the architecture
  - Can our learnings be applied to other signal processing applications?

Acknowledgements

Aliaksei Chapyzhenka, Anton Bobkov, Vladimir Pudovkin, Veronica Mikheeva, Alexey Kostyakov, Tatiana Stounina, Victoria Slavinskaya, Mariano Aguirre, Jorge Carballido, Arturo Veloz, David Arditti, Brando Perez Esparza, Victor Rivera Alvarez, Carlos Ornelas, Luis Cuellar, and Edgar Borrayo Sandoval, Jeffrey Hoffman, Thomas Tetzlaff, Frank Carroll, Kyle McCanta, Jenny Chang, Jane Lin, Kapil Gulati, David Bormann, Denise Souza, Kirk Skeba, Ernest Tsui, Inching Chen

Q&A