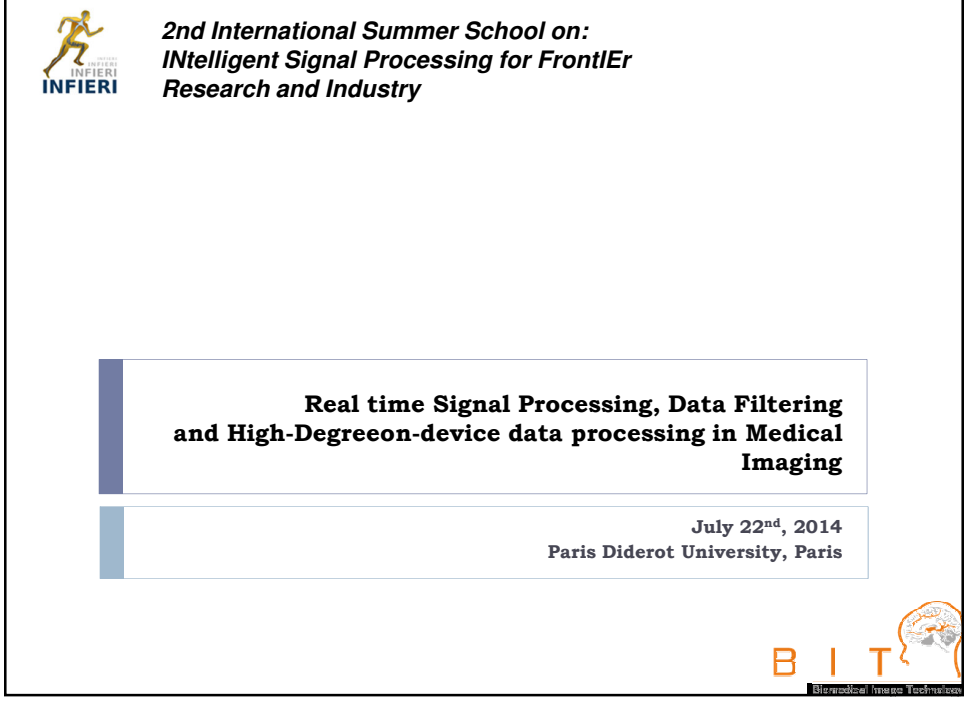
2nd International Summer School on:
INtelligent Signal Processing for FrontIEr Research and Industry

Real time Signal Processing, Data Filtering and High-Degreeon-device data processing in Medical Imaging

July 22nd, 2014
Paris Diderot University, Paris



Index

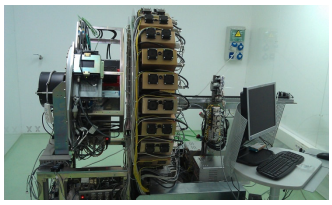
- Fixed Point Arithmetic in Signal Processing
- DSP & FPGA : Sources of processing power
- High Speed comunnications with FPGA
- SoC design with FPGA for DSP applications

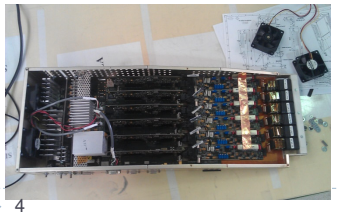
3

About me

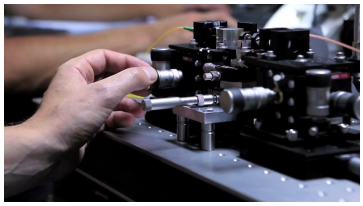
- Biomedical Image Technologies Lab @ UPM
- pedro.guerra@upm.es

B I T
Biomedical Image Technology



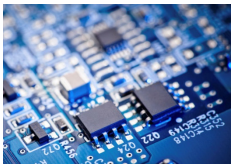


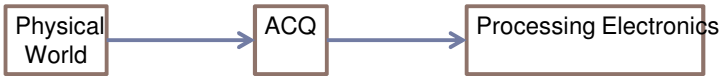
PET-CT
PET-MR
OCT



4

Algorithm





5

Algorithm

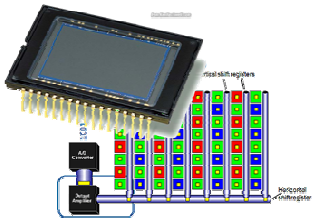
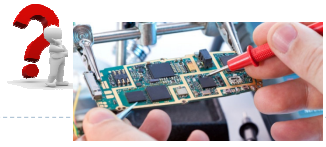


Image-processing algorithms generally fall into three classes: point operations, neighborhood operations, and more complex recursive operations.

•FPGAs are mainly used to compute point and neighborhood operations, because of their pipelined logic and on-board memory.

•More recursive operations are usually handled by an on-board/on-chip DSP



6

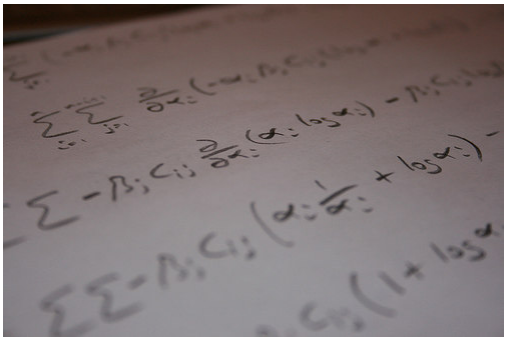
Algorithm: Implementation alternatives

Algorithm

DSP/uP
Fixed/Floating Point
Fixed Parallel Architecture
SW Development
Libraries

FPGA (+uP)
Fixed/Floating Point
Processing Architecture
FW+SW Development
IP core

7



Fixed Point Arithmetic

- Pedro Guerra Gutiérrez
- pguerra@die.upm.es

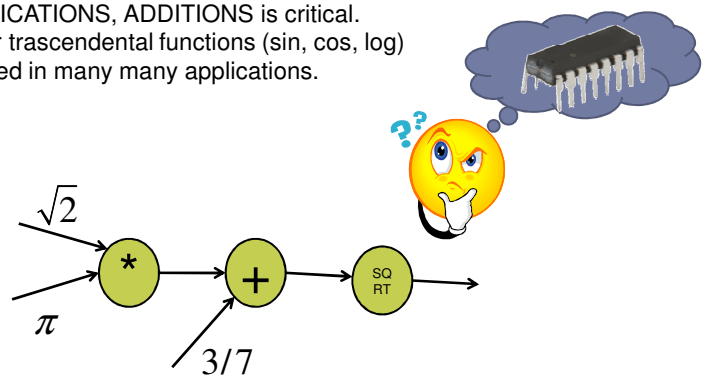
Topics

- Digital Arithmetic
  - Number representation
- Algorithm FX reduction
- Examples
  - IIR filter

9

Real Life Applications: Algorithms

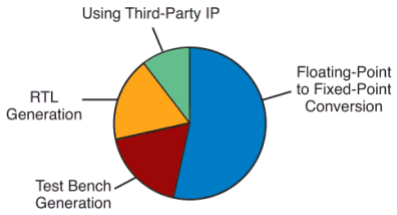
Most of digital signal processing can be seen as applied linear algebra.
The realization of operations such as MULTIPLICATIONS, ADDITIONS is critical.
Moreover trascendental functions (sin, cos, log) are needed in many many applications.



10

Fixed Point: Caveats

- FX arithmetic is complex. Then, why bothering?
  - Power/Area: floating point functions require more internal circuitry
  - Architecture: FX can be made highly parallel (real time applications)
  - Application: some application (video) blend relatively easily to FX

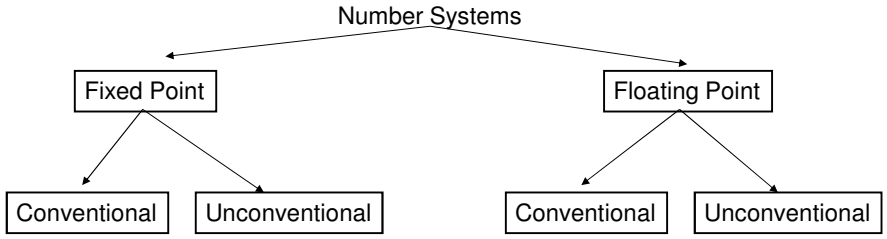


AccelChip DSP design challenges survey. www.dspdesignline.com/

11

Computer Arithmetic

- Several number representations are possible



2's Complement
1 Complement
Sign/Magnitude
Diminished by one

Logarithmic
RNS

32bit IEEE 754
64bit IEEE

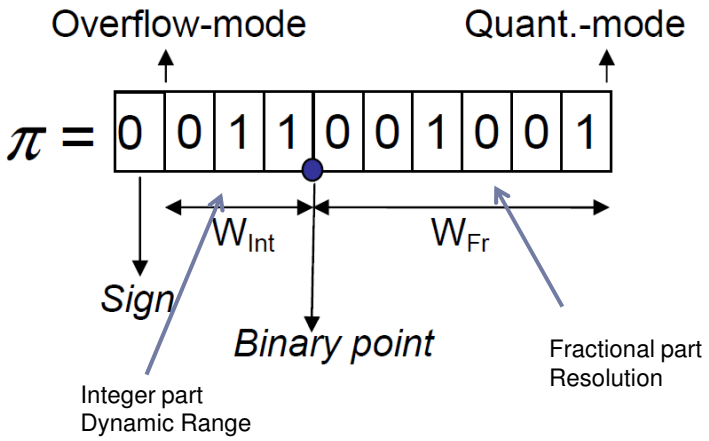
Ad-hoc
Splash II

12

Fixed Point: Caveats

- No out-of-the-box solution
- Several tools to assist designer in defining word sizes
- Application dependant
- Some operations are complex and expensive
  - Square root
  - Division (unless power of two)
  - Trigonometric
- Different architectures to implement this operations.
- Algorithm and float2fix conversion is conditioned by HW complexity of these operators.

Fixed Point Arithmetic 2'C



Fixed Point Arithmetic 2'C

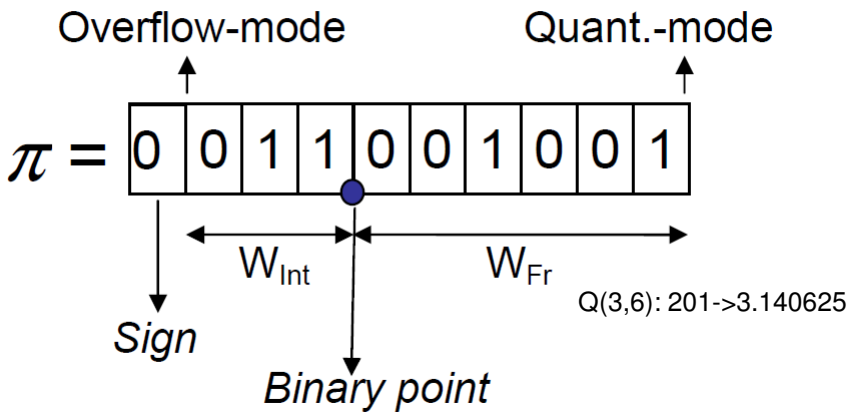
- Notation X_{FXP}: Q(n,m)
  - Fixed Point Number X_{FXP} is used to represent the real number X with (n+m+1) bits
  - X_{FXP} is in Complement 2

$$X \approx X_{FXP} \cdot 2^{-m}$$

  - Scale factors are implicit: the SW/FW designer has to keep track of them manually in the code

15

Fixed Point Arithmetic 2'C



Fixed Point Arithmetic

- **Example:** Representation of 1/7 with 8 bits

  Signed Q(0,7) :Ws=8, m+1=1, n=7:

  $1/7 \approx 18*2^{-7} = 00010011 =$

  Unsigned:Ws=8, m =0, n=8:

  $1/7 \approx 37*2^{-8} = 00100101 =$

- Whether a FX value is signed or unsigned is usually not encoded explicitly within the binary word: the sign information is implicitly defined and must be properly tracked

17

Fixed Point : Rounding

- π

  Q(2,5) = 100, E=0.016

  Q(2,5) = 101, E=0.014

  Rounding help us reduce approximation error for the same wordlength and scaling parameters.

  Comes at no cost when representing constants.

  Requires additional addition/substration for positive/negative values.

  The value to add/substract is half the smallest representable number.

  Floor(Π*2^5+0.5)=101

18

Fixed Point : Resizing / Rounding

- Sqrt(2) 1.4142135

Round/truncate

Q(2,10) => 1448 => 001.0110101000

Q(2,5) => 45 => 001.01101

=> 45 => 1.40525

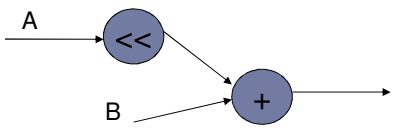
19

FPA: Addition

- Wordsize increases in one bit
- We can implement fixed point addition and subtraction by usual binary addition, provided that both operands are properly scaled

Let's say M>N

$$Y = A \cdot 2^{-N} + B \cdot 2^{-M} = \left(A \cdot 2^{M-N} + B\right) \cdot 2^{-M}$$



20

FPA: Multiplication

$$Y = A \cdot 2^{-N} \times B \cdot 2^{-M} = A \times B \cdot 2^{-(N+M)}$$

Standard integer multiplication

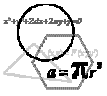
Two options for multiplication:

- Introduce saturation&rounding logic to constrain word length
- Scale numbers to [-1,1] because the product of two numbers is in the same range and keep the most significant bits. We can think of having an implied additional scale factor S.

21

FPA: Division

XILINX



$$Y = \frac{A \cdot 2^{-N}}{B \cdot 2^{-M}} = \frac{A}{B} 2^{-N+M}$$

Standard integer division

Additional documentation

SRT Division Architectures and Implementations

David L. Harris, Stuart F. Oberman, and Mark A. Horowitz

Computer Systems Laboratory
Stanford University
Stanford, CA 94305
{harrisd, oberman, horowitz}@leland.stanford.edu

22

FPA: Square Root

XILINX

$$Y = \sqrt{X \cdot 2^{-N}} = \sqrt{X}\, 2^{-\frac{N}{2}}$$

Prior scaling of X so that N is even

Additional documentation

ISCIT 2001 _____ FAA1-02

AN FPGA IMPLEMENTATION OF A FIXED-POINT SQUARE ROOT OPERATION

K. Piromsopa , C. Aporntewan and P. Chongsatitvatana

Department of Computer Engineering.
Chulalongkorn University.
254 Phayathai Road Patumwan Bangkok Thailand 10330
Phone:+66-2-218-6956, Fax:+ 66-2 -218-6955
Email:g41kpr@cp.eng.chula.ac.th , u37cap@cp.eng.chula.ac.th, prabhas@chula.ac.th

23

FPGA: Others

- The CORDIC algorithm: a unified computational scheme to perform
  - multiplication and division
  - computations of the trigonometric functions
  - computations of the hyperbolic trigonometric functions
  - exponential function, the natural logarithm and the square root

24

Fixed Point Arithmetic

- **Fundamental Rules**
  1. Bits required to represent Q(m,n): m + n + 1
  2. The range of Q(m, n) is $-2^m \le x \le 2^m - 2^{-n}$.
  3. Two binary numbers must share the scaling in order to be added
  4. $Q(m_1, n_1) + Q(m_1, n_1) = Q(m_1+1, n_1)$
  5. $Q(m_1, n_1) * Q(m_2, n_2) = Q(m_1 + m_2 + 1, n_1 + n_2)$
  6. $Q(m_1, n_1) / Q(m_2, n_2) = Q(m_1 + n_2+1 , n_1 - n_2)$
  7. $Sum(Q(m_1, n_1),i) = Q(m_1+\log_2(i), n_1)$

25

Fixed Point Arithmetic: Examples

$$\frac{\pi \cdot 0.4573}{-33.\widehat{3}} = -0.0431$$

$$\frac{\pi \cdot 0.4573}{-33.\widehat{3}} = -0.0431$$

$$\frac{\pi\big|_{(2,5)} \cdot 0.4573\big|_{(0,7)}}{-33.\widehat{3}\big|_{(6,1)}}$$

26

Fixed Point Arithmetic: Examples

$$\frac{\pi \cdot 0.4573}{-33.\widehat{3}} = -0.0431$$

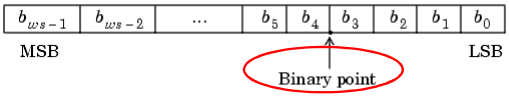
$$\frac{\pi|_{(2,5)} \cdot 0.4573|_{(0,7)}}{-33.\widehat{3}|_{(6,1)}} = \frac{101|_{(2,5)} \cdot 59|_{(0,7)}}{-67|_{(6,1)}} = \frac{5959|_{(3,12)}}{-67|_{(6,1)}} = -89|_{(5,11)} = -0.0435$$

27

Fixed Point Arithmetic

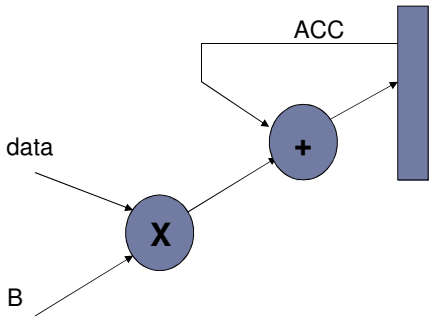
- Unconstrained, wordlength grows indefinitvely.
- Every now and then, we need to limit wordlength.
  - Saturation (restrict dynamic range)
  - Rounding (restric accuracy)

Q(ws-1-4,4)



28

FXP Arithmetic : Example



29

Fixed Point Arithmetic: Examples

$$\frac{\sqrt{\pi}\cdot\sqrt{2}}{1/7} = 17.5464$$

$$\frac{\sqrt{\pi}\big|_{(2,5)}\cdot\sqrt{2}\big|_{(1,6)}}{1/7\big|_{(0,7)}}$$

$$\sin\left(\frac{\pi}{4}\right)\left\{\begin{array}{l}\frac{\pi}{4} = 25\cdot 2^{-5}\\ \frac{\pi}{4} = 16\cdot 2^{-6}\cdot\pi\end{array}\right\} = \left\{\begin{array}{l}90\cdot 2^{-7}\\ 91\cdot 2^{-7}\end{array}\right.$$

$$\frac{\sqrt{\pi}\cdot\sqrt{2}}{1/7} + \sin\left(\frac{\pi}{4}\right) = 18.2535$$

30

Fixed Point Arithmetic: Examples

$$\frac{\sqrt{\pi}\cdot\sqrt{2}}{1/7}=17.5464_{\square}$$

$$\frac{\sqrt{\pi}\big|_{(2,5)}\cdot\sqrt{2}\big|_{(1,6)}}{1/7\big|_{(0,7)}}=\frac{57\big|_{(2,5)}\cdot 91\big|_{(1,6)}}{18\big|_{(0,7)}}=\frac{5187\big|_{(4,11)}}{18\big|_{(0,7)}}=288\big|_{(12,4)}=18.0_{\square}$$

$$\sin\left(\frac{\pi}{4}\right)\begin{Bmatrix}\frac{\pi}{4}=25\cdot 2^{-5}\\ \frac{\pi}{4}=16\cdot 2^{-6}\cdot\pi\end{Bmatrix}=\begin{Bmatrix}90\cdot 2^{-7}\\ 91\cdot 2^{-7}\end{Bmatrix}$$

$$\frac{\sqrt{\pi}\cdot\sqrt{2}}{1/7}+\sin\left(\frac{\pi}{4}\right)=18.2535_{\square}$$

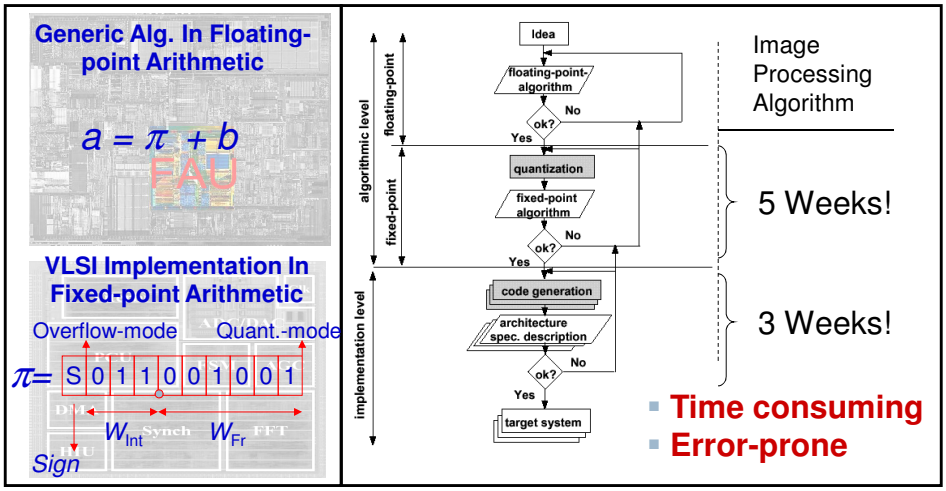
$$288\big|_{(12,4)}+90_{(1,7)}=2304\big|_{(12,7)}+91_{(1,7)}=2394_{(13,7)}=18.71_{\square}$$

31

- 0.0000000057?

32

Algorithfm: FP to FX conversion

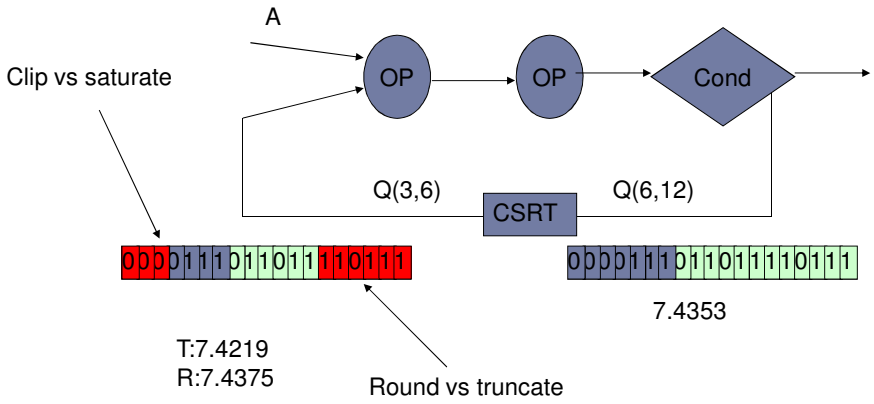


33

FP to FX conversion

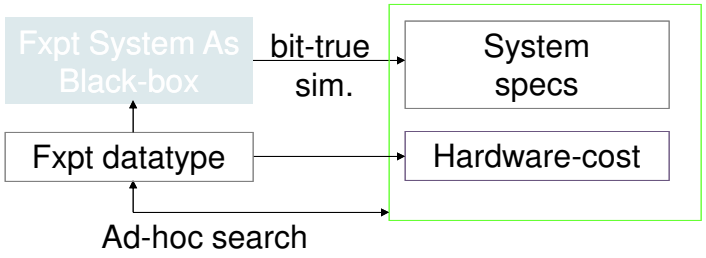
- Complex algorithms are not easily ported from FP to FX.
- Wordlengths are precision must be determined for **every variable** and tracked along the different loops and paths of the algorithms.
- Wordlengths would probably grow limitless in loops unless saturation and rounding units (CSRT) where placed at proper points to constrain word sizes.
- Wordlength fixation (clip/saturation & round/truncation) introduces noise that degrades the algorithm behavior.
- Usually **specs/figures of merit** are available to determine the maximum tolerable degradation.

34



35

Robust Adhoc



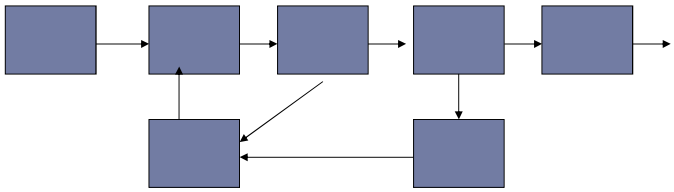
- ✓ Long bit-true simulation
- ✓ Large number of iterations
- ✓ Tedious for large systems

Wonyong Sung, and Ki-Il Kum, "Simulation-based word-length optimization method for fixed-point digital signal processing systems," IEEE Trans. Signal Processing, vol. 43, no. 12, Dec. 1995.

36

Robust Adhoc

Complex Algorithm with many feedback loops

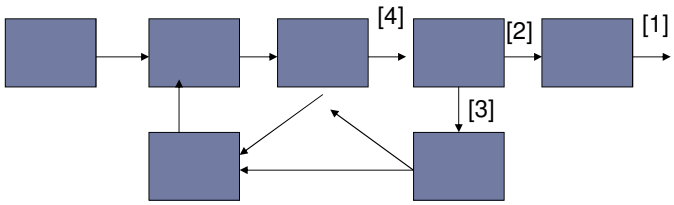


Break the loop
Determine minimum wordlengths and accuracy at high level block, starting from the end towards the beggining

37

Robust Adhoc

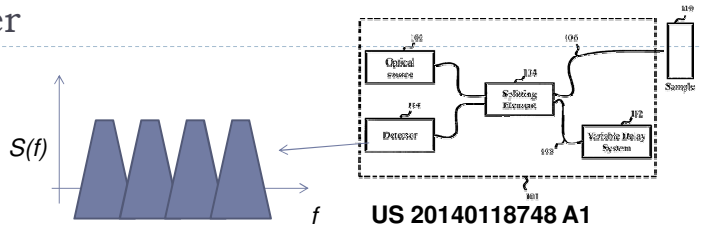
Complex Algorithm with many feedback loops

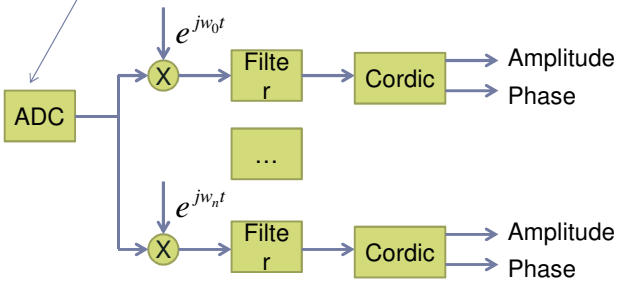


Break the loop: divide and conquer
Determine minimum wordlengths and accuracy at high level block, starting from the end towards the beginning
May result in suboptimum implementation, but will enable a working system

38

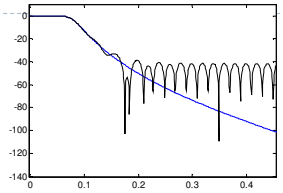
Application Example: OCT Quadrature receiver





39

Application Example: OCT Quadrature receiver



- Constraints
  - Sharp lowpass filter
    - IIR order 6
    - FIR: order 90!!!
  - MUL Operations shall fit in a MAC module
    - Need quantization tolerant architecture
  - Avoid complex feedback loops
    - Must sustain 80 MSPS: Select appropriate architecture

40

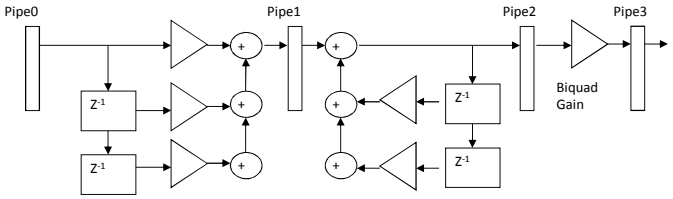
Application Example: OCT Quadrature receiver

- Possible IIR implementations
  - Direct form I and direct form II
  - Cascade
  - Parallel
  - Lattice canonical forms
- **Direct form I** : often used by fixed-point IIR filters since a larger single adder can be used to prevent saturation.
- **Direct form II** : often used by floating-point IIR filters, since it uses fewer states and the adders are not as sensitive to saturation.
- **Cascade canonical form:** has the lowest quantization sensitivity, but at the cost of additional resources.

41

Application Example: OCT Quadrature receiver

IIR building block: Biquad 2nd order



Resources per biquad	Number
Tables	0
Multipliers	6
Adder	4
Registers	8

IIR :divided into biquadratic sections (SOS), which are which are robust to quantization. Each section is implemented as DF-I
The architecture eliminates feedback loops and enables **pipelining**.
The number of wordsizes/scale factors is small.

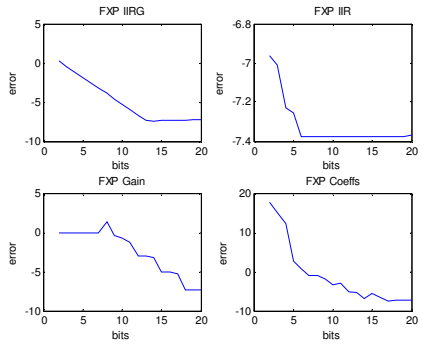
Synthesis	Slice Logic Utilization:	
	Number of Slice Registers:	9611 out of 269200 3%
	Number of Slice LUTs:	10699 out of 134600 7%
	Number of Block RAM/FIFO:	4 out of 365 1%
	Number of DSP48E1s:	**200 out of 740 27%**
Timing	Minimum period: 12.438ns{1}	**Maximum frequency: 80.399MHz**

42

Application Example: OCT Quadrature receiver

Attention is paid to avoid IIR filter instabilities due to changes in pole location of the quantized filter.
In the analysis it is assumed that the ADC input signal has zero mean and it is represented as a Q(0,13): 14 bit signed value
Wordsize is optimized for a order 6 IIR filter with SOS considering impulse response

Representation of GAIN coefficients Representation of IIR output

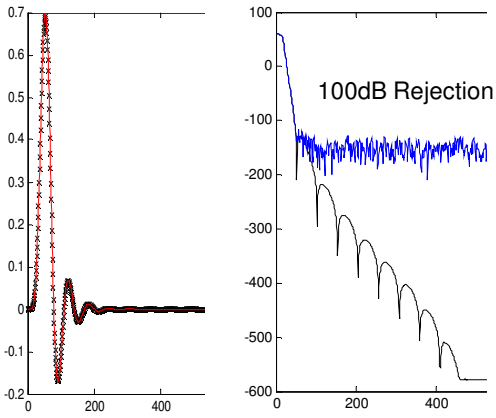


Representation of intermediate values Representation of IIR coefficients

43

Application Example: OCT Quadrature receiver

Float (Matlab) vs Fixed (VHDL/modelsim) impulse response



Parameterized IP core is generated for the IIR filter.

Matlab script automatically generates constant files for any filter order

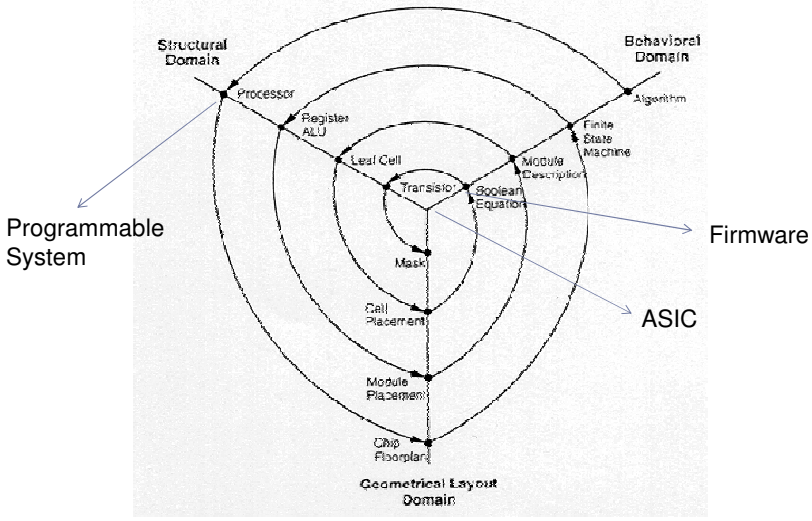
44

HLS Tools

- HDL still useful
- Architecture exploration
- New laguage/tools for HLS becoming available
  - Handel-C (now defunct)
  - SystemC
  - Vivado
- Overloaded types to constrain wordlength

45

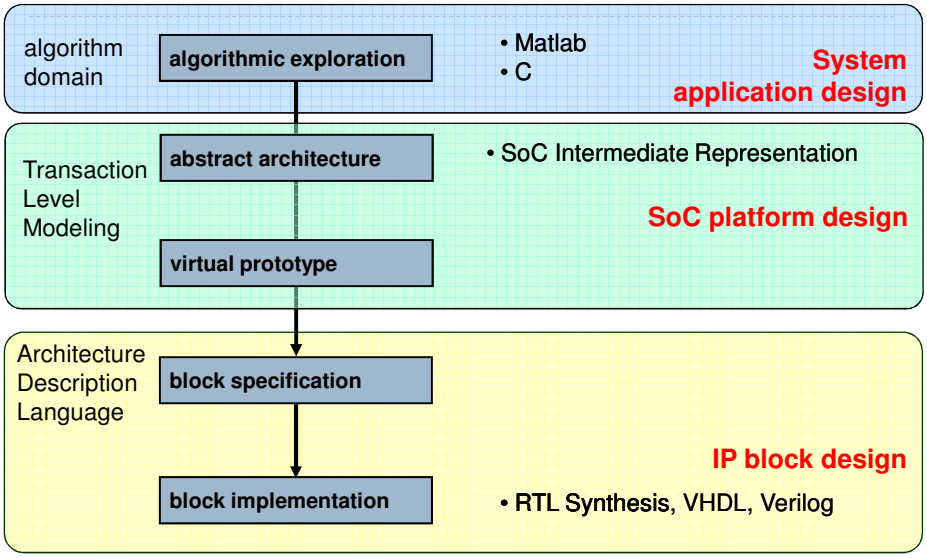
HLS Tools



46

Gajski-Kuhn chart

Refinement : from algorithm to hardware



47

Takeaways

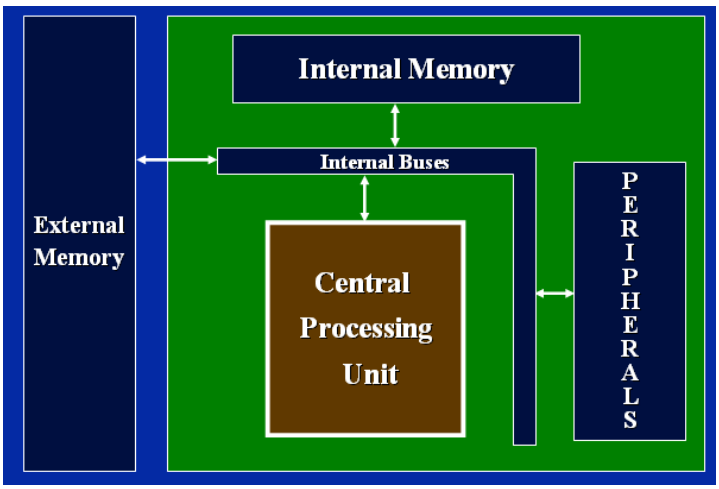
- The real-time (RT) implementation of most signal processing algorithms include a floating to fixed point conversion phase.
- This phase is time consuming and may require algorithm modifications or simplifications.
- RT constraints impose constraints on the processing architecture: area/power/clock speed
- Image processing problems may be solved with DSP/uP/FPGA. The actual technology depends on the constraints and skills.

48

DSP

TI's C5000
Architecture Overview

General DSP System Block Diagram



50

Common Architectural Features of DSPs

Optimized for Data crunching:

- ✓MAC- multiply/ accumulate Unit
  - ✓ Executes one or more MAC in one clock cycle
- ✓Internal registers for storing operands and results
- ✓Multiple internal data memories
- ✓Auxilary adressing registers (8 or more) with a dedicated ALU
- ✓Multiple buses and dual access to data memory
- ✓Separate access to program and data memories:
  - ✓ (Harvard architecture)
- ✓DMA (Direct Memory Access)
- ✓Some DSP have floating point units

51

TMS320C54X Key Features

Advanced multibus architecture with:
- ❖ one **program** bus, three **data** buses, and four **address** buses
- ❖ 40-bit arithmetic logic unit (**ALU**), including a 40-bit barrel shifter and two 40-bit accumulators
- ❖ 17-bit × 17-bit parallel multiplier coupled to a 40-bit dedicated adder non-pipelined single-cycle multiply/accumulate (**MAC**) operation
- ❖ Compare, select, store unit (**CSSU**) for the Viterbi operator implementation
- ❖Two address generators, including eight auxiliary registers (ARn) and two auxiliary register arithmetic units (ARu)

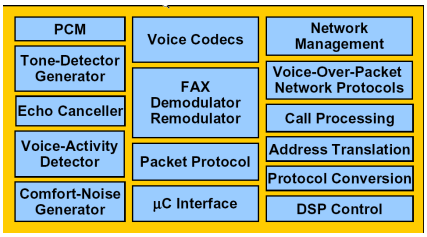
52

Comparison Between C54x and C55x

	C54x	C55x
MACs	1	2
Accumulators	2	4
Read buses	2	3
Write buses	1	2
Program fetch	1	1
Address buses	4	6
Program word size	16 bits	8/16/24/32/40/48 bits
Data word size	16 bits	16 bits
Auxiliary Register ALUs	2 (16-bit each)	3 (24-bit each)
ALU	1 (40-bit)	1 (40-bit)/1 (16-bit)
Auxiliary Registers	8	8
Data Registers	0	4
Memory	separate data/prog	Unified space

Common Architectural Features of DSPs

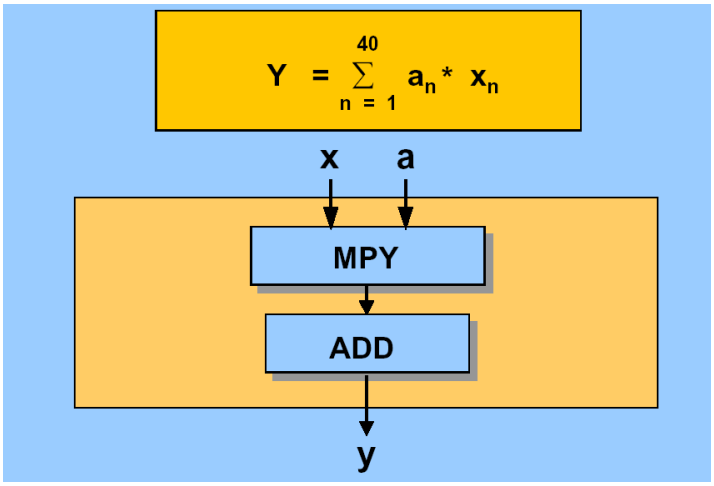
❑Real-time and Multi-tasking:
- ✓Hardware and software interrupt handling
- ✓Zero overhead loops
- ✓Pipelined instructions
- ✓Parallel units
- ✓Specialised peripherals
- ✓RT-OS available



❑Complex Algorithms:
- ✓Specialised instructions (Viterbi, LMS…)
- ✓Bit reverse adressing (FFT)
- ✓Circular buffers (FIR filters)

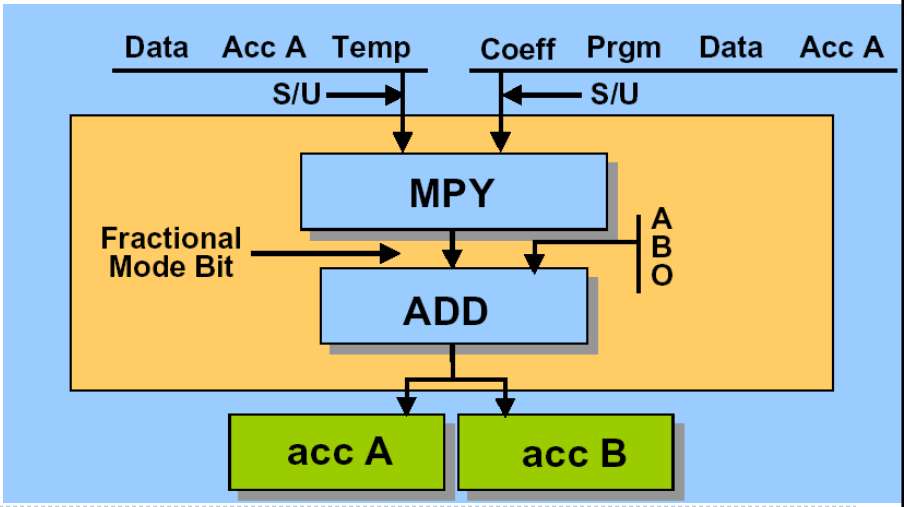
54

DSP Requires Multiply and Accumulate



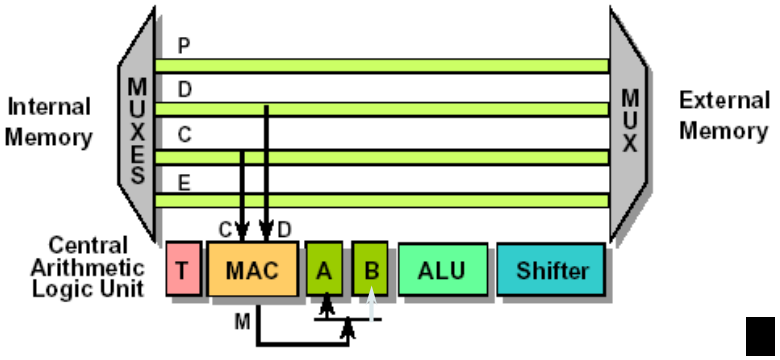
55

Multiply and Accumulate Unit



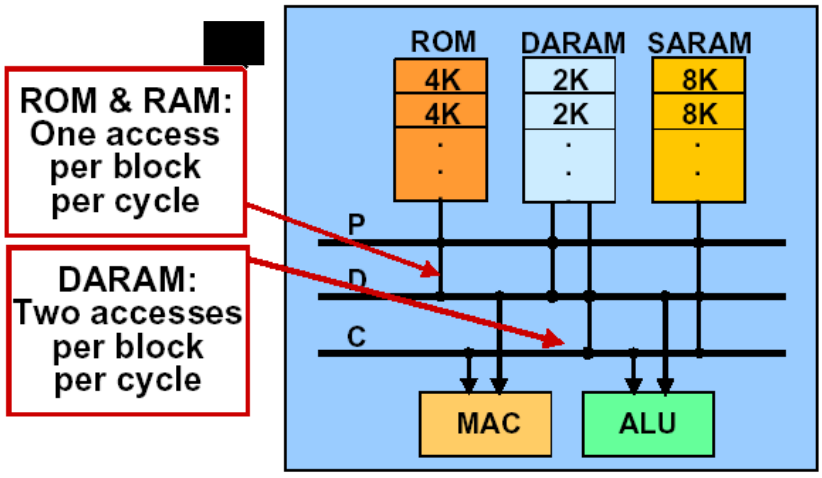
56

Multiple Buses



57

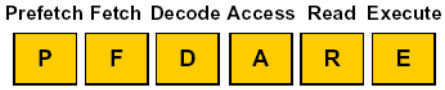
Internal Memory for Fast Access



58

Instruction Pipeline for Fast Execution

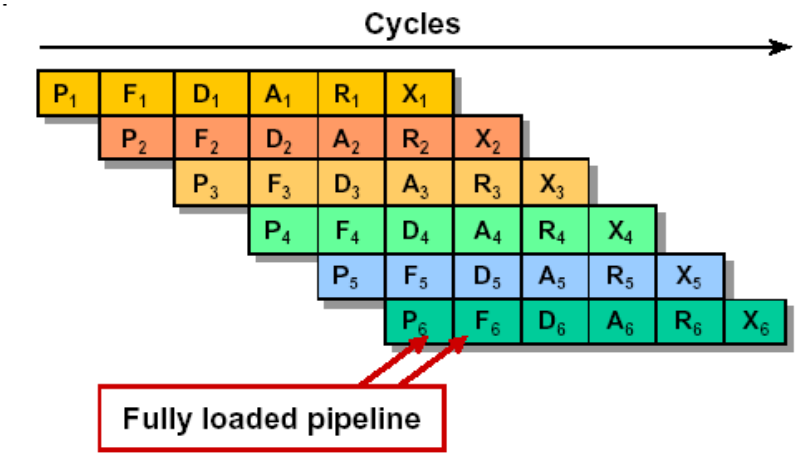
Instruction is broken by the compiler into smaller tasks that can be executed in parallel



- Prefetch: Calculate address of instruction
- Fetch: Collect instruction
- Decode: Interpret instruction
- Access: Collect address of operand
- Read: Collect operand
- Execute: Perform operation

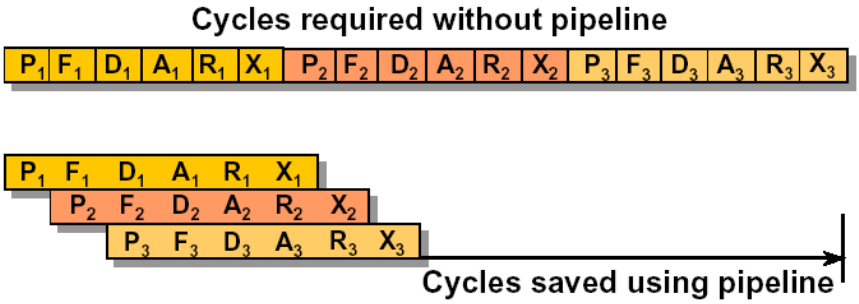
59

Sequential Processing of Instructions



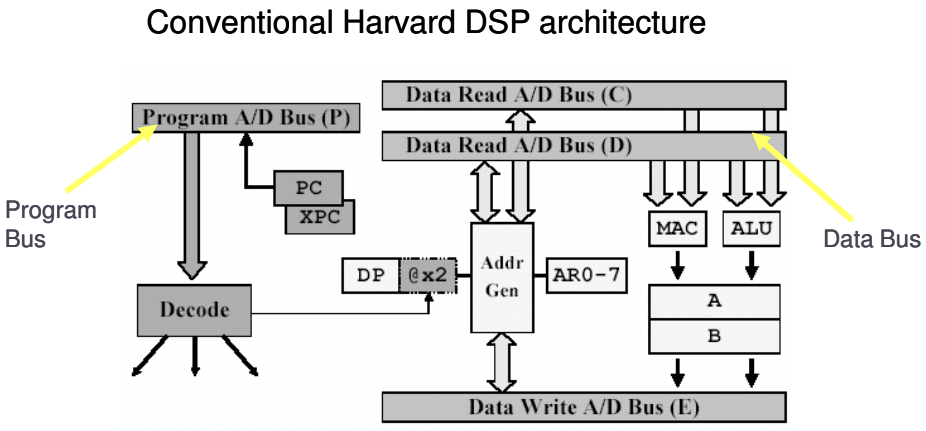
60

Less Cycles per Instruction
Less Power Consumption



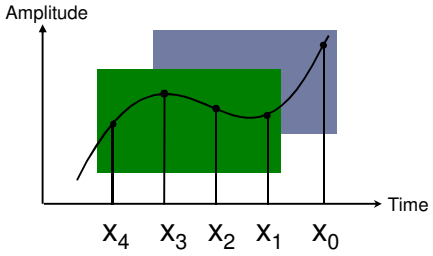
61

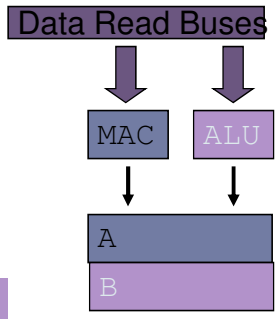
TMS320C54X DSP Generation Block Diagram



62

Convolution with the C54x 1 of 2





$$y_0 \leftarrow \sum_{n=0}^{3} a_n x_n$$

$z = x_2 + x_4 + x_3 + x_1$

- Single-cycle MAC
- Single-cycle ADD

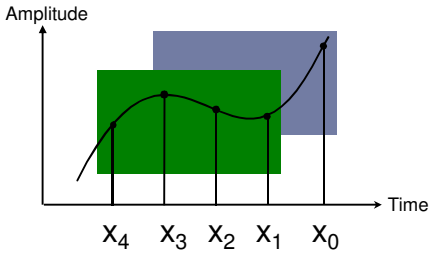
MAC *AR2+, *AR3+, A

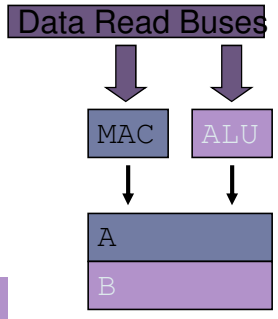
ADD @x2, B ...

63

3

Convolution with the C54x 2 of 2





$$y_0 \leftarrow \sum_{n=0}^{3} a_n x_n$$

$z = x_2 + x_4 + x_3 + x_1$

- Single-cycle MAC
- Single-cycle ADD

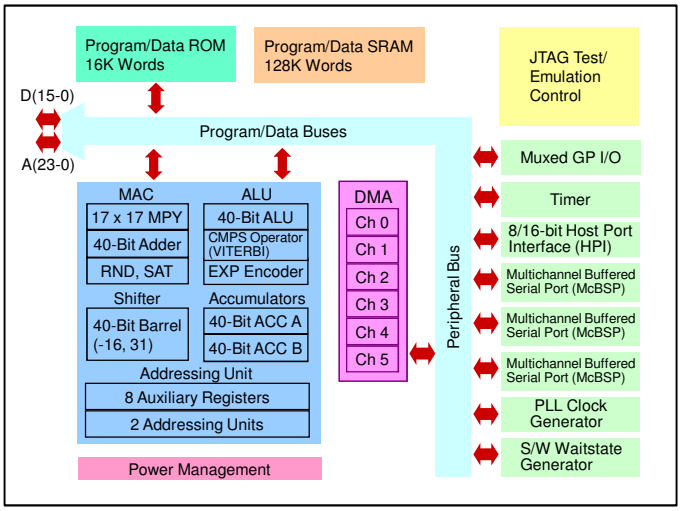
MAC *AR2+, *AR3+, A

ADD @x2, B ...

64

3

C54x Block Diagram



- 17x17 MAC Unit
- Saturation and Rounding Hardware
- Two 40-bit ACC's
- 40-bit ALU
- 40-bit Barrel Shifter
- Temporary Register
- Exponent Encoder
- Program and Data Address Generation Units
- Compare, Select and Store Unit
- 4 Internal Bus Pairs
- External Interface

C5416 example

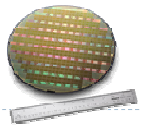
65

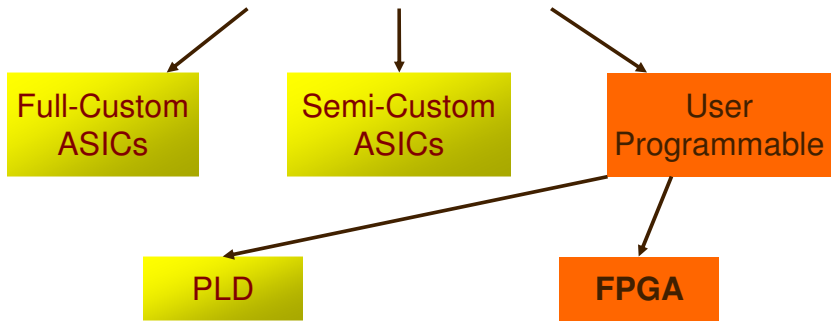
5

FPGA

Architecture Overview
Xilinx Series 7

IC Overview

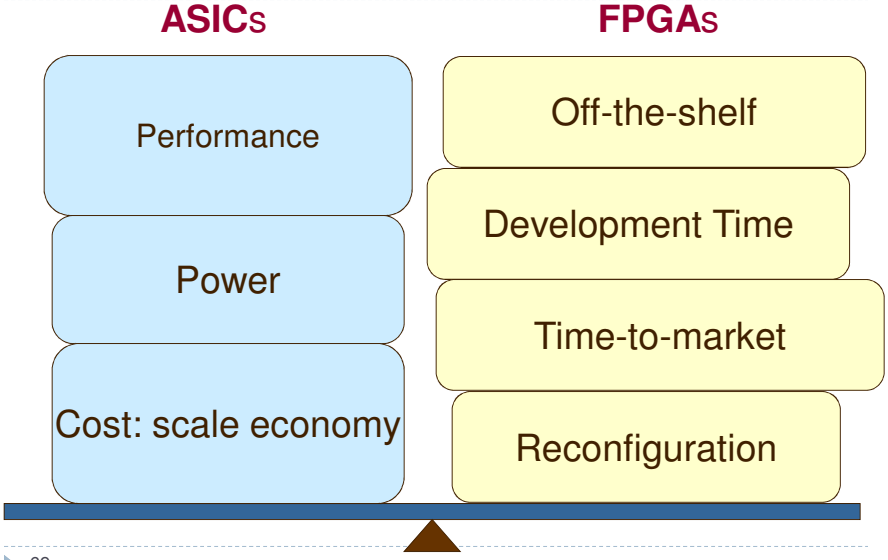




IC Overview

- An **application-specific integrated circuit** (ASIC) is an integrated circuit (IC) customized for a particular use .
- As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million.
- Modern ASIC often include entire processors (ARM), memory blocks including ROM, RAM, and other large building blocks, such as ethernet interfaces.
- Such an ASIC is often termed a SoC (System-on-a-chip). Designers of digital ASICs use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality and implement and ASICs.

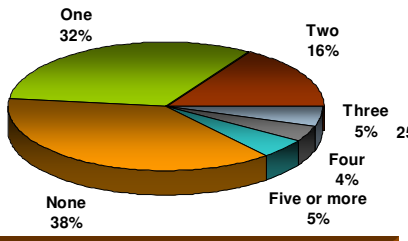
Which option shall we choose?



69

System Design Trends

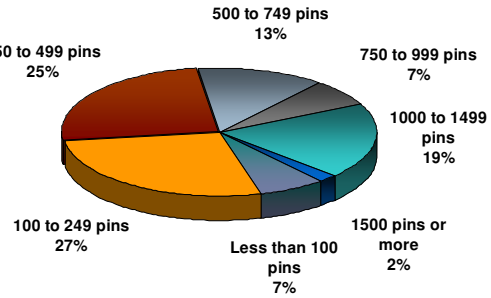
Average FPGAs on a single PCB



Almost half of all PCBs with FPGAs have multiple FPGAs per PCB

CMP
EETIMES
2004 EDA Study

Average pins of largest FPGA



Many FPGAs now over 500 pins

From "Solutions Using FPGAs", Dan Gardner

70

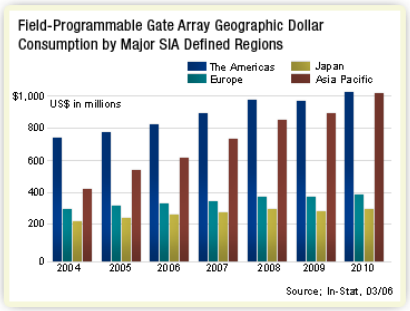
Main FPGA manufacturers

FPGAs based on SRAM

- **Xilinx, Inc.**
- **Altera Corp.**
- Atmel
- Lattice Semiconductor

FPGAs based on Flash & antifuse

- Actel Corp.
- Quick Logic Corp.

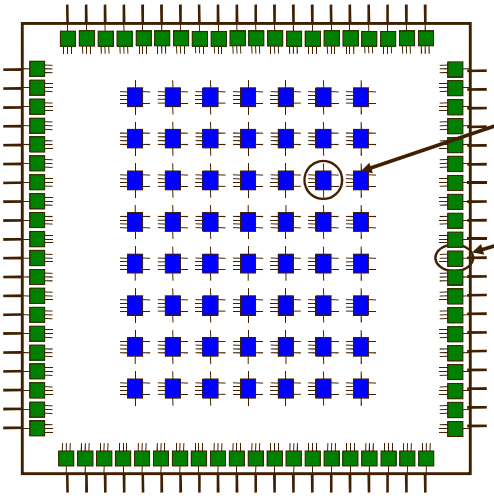


Platform FPGA Technology vs. Pushbutton FPGA Design Flow

- Major technology changes in platform FPGA
  - More gates per part
  - Specialized embedded technology
  - Specialized interconnect
  - High-speed device effects
- Emerging design challenges
  - Design debug process is unreliable for complex designs
  - Harder to bring quality product to market quickly
  - Harder to manage flows: Design reuse & IP integration
  - Harder to predict development costs

From "Solutions Using FPGAs", Dan Gardner

What is an FPGA?



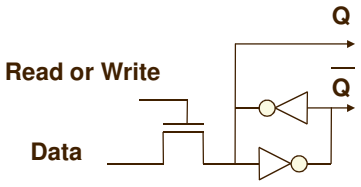
Configurable Unit

I/O Blocks

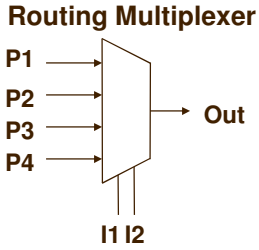
Interconnetion matrix

Basic Tecnology
Function implementation
Combinational/Sequential Logic
IO connectivity

What is an FPGA?: SRAM bit cell



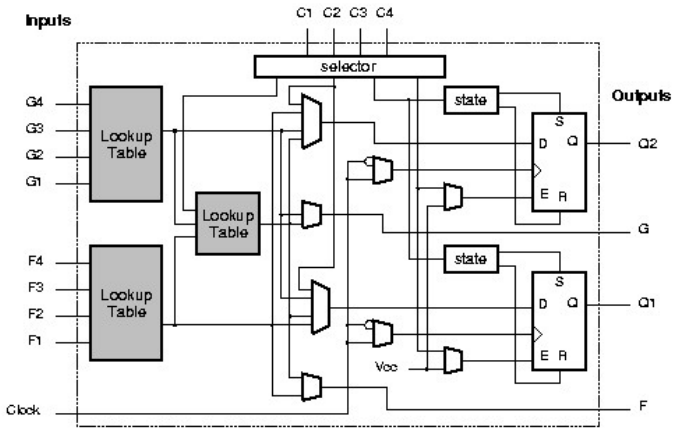
Programming Bit



2-bits

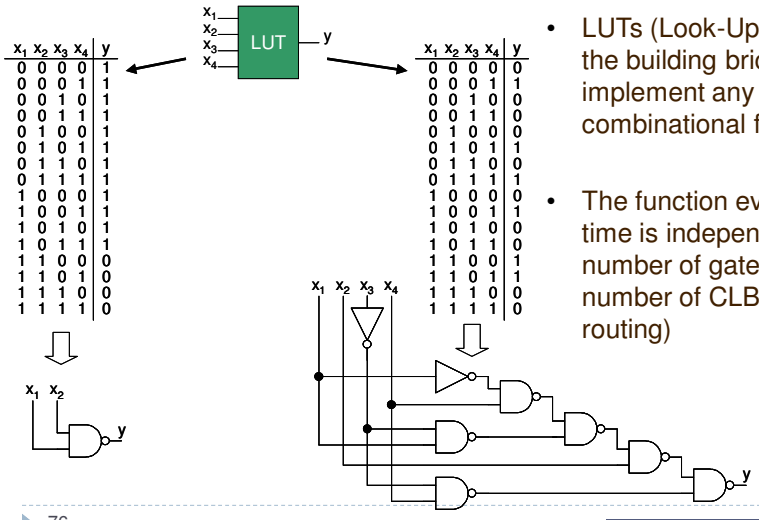
- 5/6 transistors memory cell
- Static random access memory (SRAM) can retain its stored information as long as power is supplied.
- The structure of a 5/6 transistor SRAM cell, storing one bit of information, can be seen in Figure
- The core of the cell is formed by two CMOS inverters, where the output potential of each inverter is fed as input into the other . This feedback loop stabilizes the inverters to their respective state.

What is an FPGA?: Basic Programming Cell



Xilinx XC4000 CLB

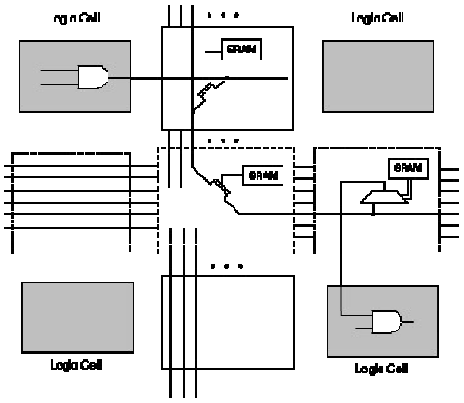
What is an FPGA?: Basic Programming Cell



- LUTs (Look-Up tables) are the building brick to implement any combinational function.
- The function evaluation time is independent of the number of gates, but the number of CLB (and routing)

What is an FPGA?: Basic Routing Matrix

- SRAM bit to connect vertical lanes and horizontal lanes



Xilinx XC4000 CLB

2nd International Summer School on:
INtelligent Signal Processing for FrontIEr Research and Industry

Architecture of modern FPGAs

Xilinx 7 Series FPGA Overview

XILINX

Example: Xilinx-7 Series FPGA Families

Maximum Capability	ARTIX-7 Power and Cost	KINTEX-7 Price/Performance	VIRTEX-7 System Performance
Logic Cells	20K – 355K	70K – 480K	285K – 2,000K
Block RAM *	12 Mb	34 Mb	65 Mb
DSP Slices *	40 – 700	240 – 1,920	700 – 3,960
Peak DSP Perf.	504 GMACS	2,450 GMACs	5,053 GMACS
Transceivers *	4	32	88
Transceiver Performance	3.75Gbps	6.6Gbps and 12.5Gbps	12.5Gbps, 13.1Gbps and 28Gbps
Memory Performance	1066Mbps	1866Mbps	1866Mbps
I/O Pins	450	500	1,200
I/O Voltages	3.3V and below	3.3V and below 1.8V and below	3.3V and below 1.8V and below

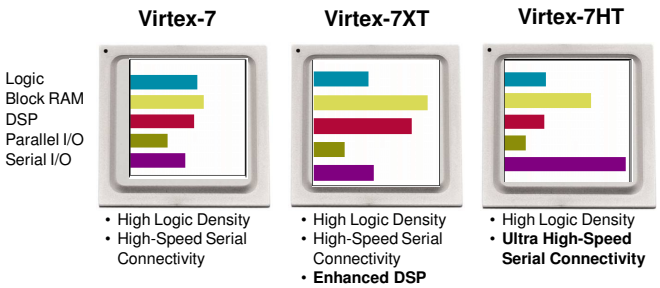
79

Architecture of modern FPGA

XILINX

Virtex-7 Devices

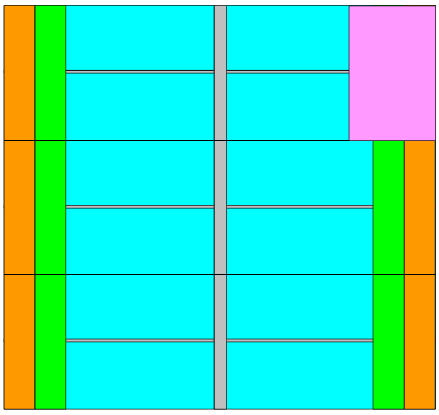
- The Virtex-7 family has several devices
  - Virtex-7: General logic
  - Virtex-7XT: Rich DSP and block RAM, higher serial bandwidth
  - Virtex-7HT: Highest serial bandwidth



80

7 Series FPGA Layout

- All devices contain two I/O columns
  - Contains parallel I/O resources
- Clock Management Tile (CMT) columns are adjacent to I/O columns
  - Enables high speed I/O interfaces
- Clock routing resources are in the center column
- High-speed serial I/O replace I/O banks in smaller devices or are contained in additional columns in larger devices



81

Architecture of modern FPGA

XILINX

Architecture Alignment

- Common elements enable IP reuse for quick design portability across all 7 series families
  - Design scalability from low-cost to high-performance

Logic Fabric
LUT-6 CLB

Precise, Low Jitter Clocking
MMCMs

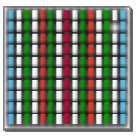
On-Chip Memory
36Kbit/18Kbit Block RAM

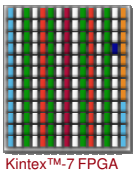
Enhanced Connectivity
PCIe® Interface Blocks

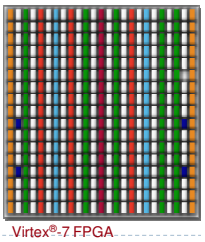
DSP Engines
DSP48E1 Slices

Hi-perf. Parallel I/O Connectivity
SelectIO™ Technology

Hi-performance Serial I/O Connectivity
Transceiver Technology


Artix™-7 FPGA


Kintex™-7 FPGA

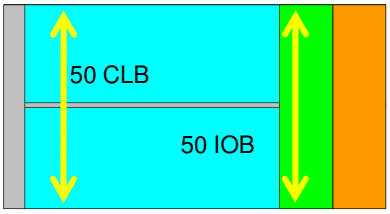

Virtex®-7 FPGA

82

Architecture of modern FPGA

Clock Regions and I/O Banks

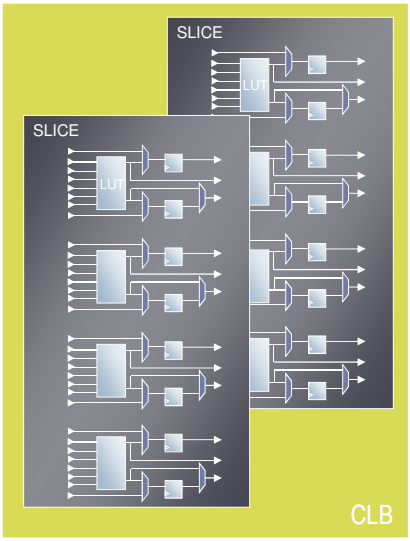
- Each clock region is 50 CLBs tall
  - An increase from 40 CLBs in previous technologies
  - Regional clock resources remain in the center of the clock region
    - 25 rows of CLBs above and below the clock routing
- I/O banks are 50 IOBs tall
  - An increase from 40 IOBs in previous technologies
  - I/O banks and clock regions are aligned, like in previous technologies



83

Processing Blocks: CLB Structure

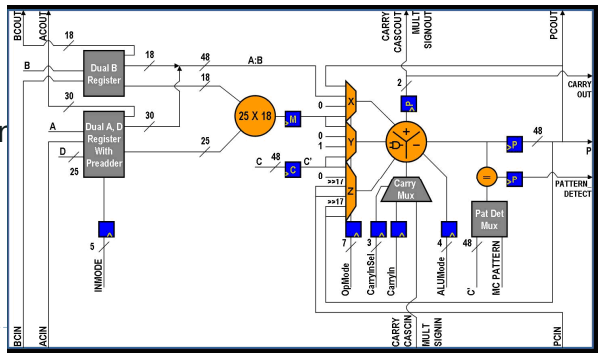
- Two side-by-side slices per CLB
  - Slice_M are memory-capable
  - Slice_L are logic and carry only
- Four 6-input LUTs per slice
  - Consistent with previous architectures
  - Single LUT in Slice_M can be a 32-bit shift register or 64 x 1 RAM
- Two flip-flops per LUT
  - Excellent for heavily pipelined designs



84

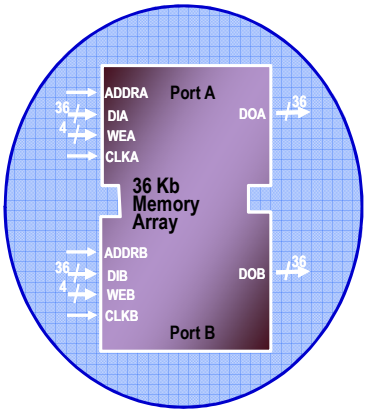
Processing Blocks: DSP Slice

- Integrates multiple functionalities
  - 25x18 multiplier
  - 25-bit pre-adder
  - Flexible pipeline
  - 96-bit MACC
  - 48-bit ALU
  - Pattern detect
  - 17-bit shifter
  - Dynamic operation (cycle by cycle)



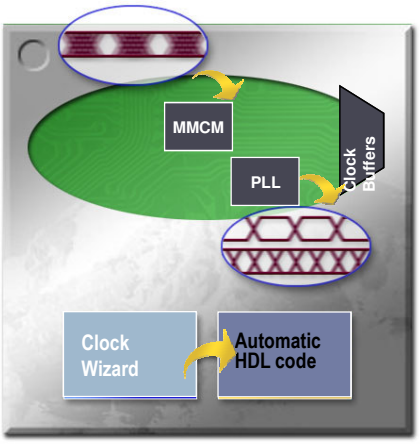
Processing Blocks: Block RAM

- 36K/18K block RAM
- Close to DSP slices for eficient implementation
- Configurations (same as Virtex-6 FPG)
  - 32k x 1 to 512 x 72 in one 36K block
  - Simple dual-port and true dual-port configurations
  - Built-in FIFO logic
  - 64-bit error correction coding per 36K block
  - Adjacent blocks combine to 64K x 1 without extra logic



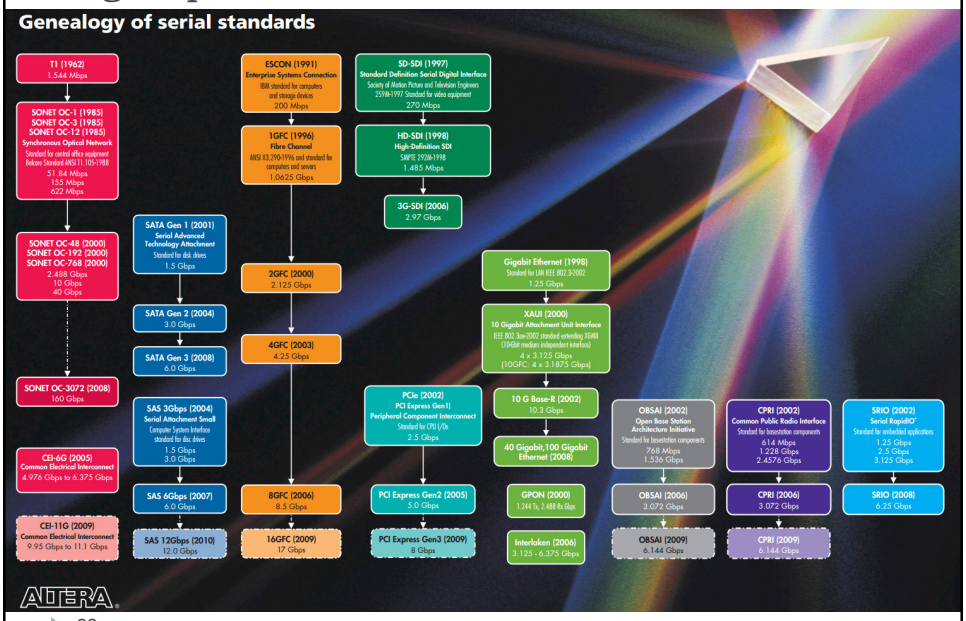
Clocking Resources

- Based on the established Virtex-6 FPGA clocking structure
- Low-skew clock distribution
  - Combination of paths for driving clock signals to and from different locations
- Clock buffers
  - High fanout buffers for connecting clock signals to the various routing resources
- Clock regions
  - Device divided into clock regions with dedicated resources
- Clock management tile (CMT)
  - One MMCM and one PLL per CMT
  - Up to 24 CMTs per device



87

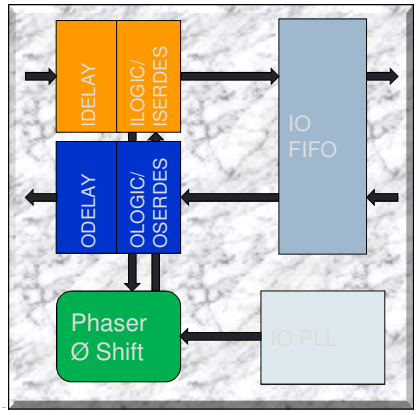
High-Speed Serial Standards



88

Input/Output Blocks

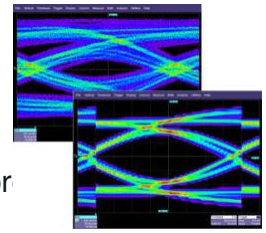
- Two distinct I/O types
  - High range: Supports standards up to 3.3V
  - Higher performance with more I/O delay capability
- Extension of logic layer functionality
  - Wider input/output SERDES
  - Addition of independent ODELAY
- New hardware blocks to address highest I/O performance
  - Phaser, IO FIFO, IO PLL



89

High-Speed Serial I/O Transceivers

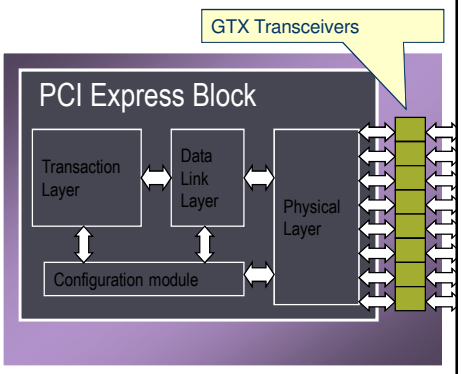
- GTP transceivers – up to 3.75 Gbps
  - Ultra high volume transceiver
  - Wire bond package capable
- GTX transceivers – up to 12.5 Gbps
  - Support for the most common 10 Gbps pr
- GTH transceivers – up to 13.1 Gbps
  - Support for 10 Gbps protocols with high FEC overhead
- GTZ transceivers – up to 28 Gbps
  - Enables next generation 100–400Gbps system line cards



90

PCI Express

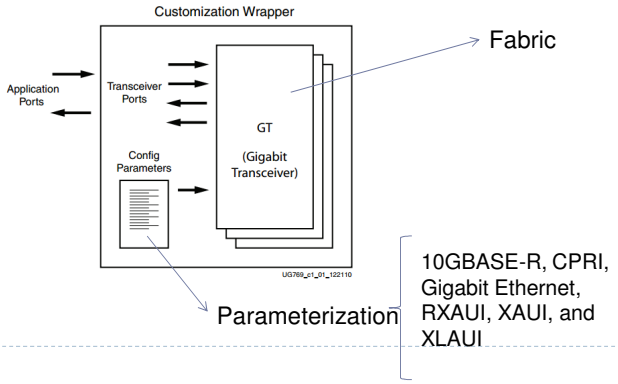
- Features
  - Compliant to PCIe Revision 2.1
  - Endpoint & root port
  - AXI user interface
  - <100 ms configuration*
  - FPGA configuration over PCI Express*
  - End-to-end CRC*
  - Advanced error reporting*
  - 100-MHz clocking
- New wrappers
  - Multi-function*
  - Single-root I/O virtualization*
- Configurations
  - Lane widths: x1-8
  - Data rates: Gen1 & Gen2 (2.5/5.0 Gbps)
  - Dependent on GT and fabric speed



91

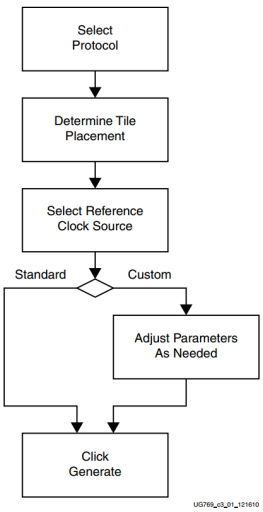
High-Speed interfacing

- Menu driven interface allows transceivers to be configured for popular industry standards, producing a wrapper for quick integration.



92

High-Speed physical interfacing

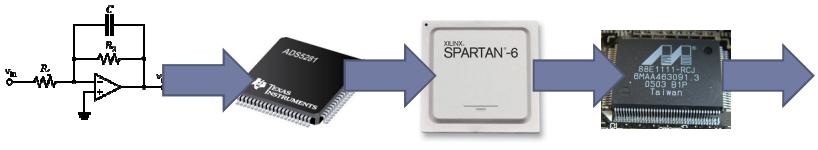


- Wizards allow handling physical design constraints, as well as RX/TX clocking.
- We end up with a FIFO like interface, with is easily integrated

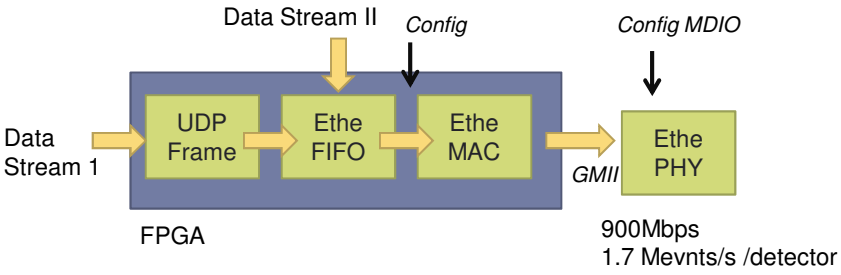
93

Application Example: Ethernet data streaming

Acquisition electronics in preclinical PET



FPGA GbE PHY



94

Application Example: Ethernet data streaming

- *Unreliable* – When a message is sent, it could get lost along the way:
- *Not ordered* – If two messages are sent to the same recipient, the order in which they arrive cannot be predicted.
- *Lightweight* – There is no concept of acknowledgment, retransmission, or timeout.
- *Datagrams* – Packets are sent individually and are checked for integrity only if they arrive.
- The HW UDP creates data frames from stream data
  - Fixed UDP configuration: fragmentation
  - Self-content data with numbered frames (70 single events/frame)
- Module inserts headers and CRC in real time.
  - 900 Mbps measure on GbE link
- New frame started when enough data is waiting in the input queue (or timeout)
- The module reuses IP module for MAC
- TCP/IP data retrieval in the PC is easy

95

PHY Chip Architecture

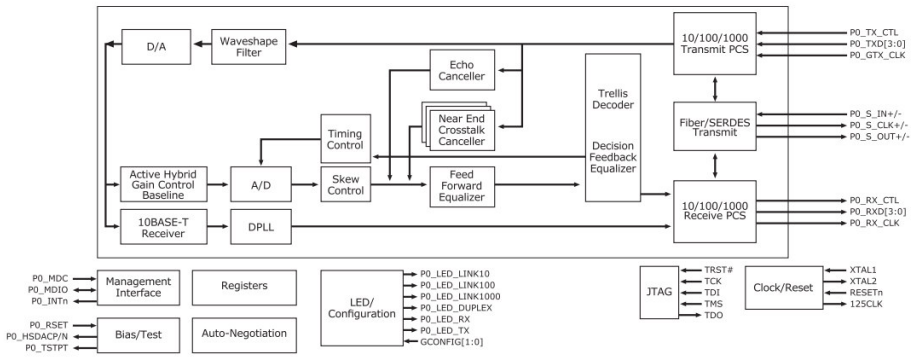
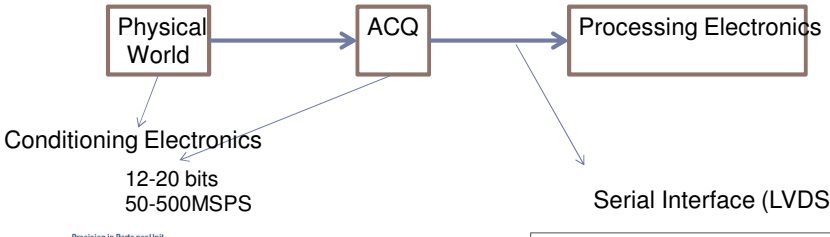
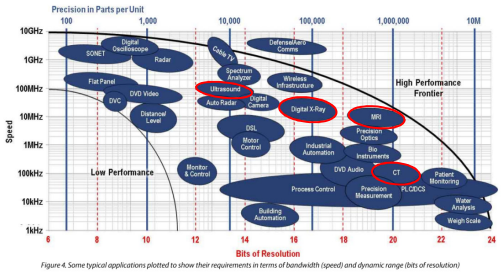


Fig 1. Alaska Single-Port GbE Transceiver (88E1111) Block Diagram

96

High-Speed interfacing: ADC front-end





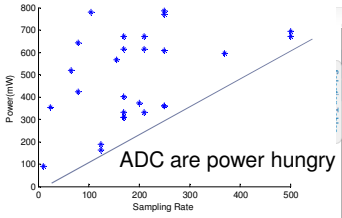
Typical applications and requirements.
Source: Analog Devices Article MS-2629

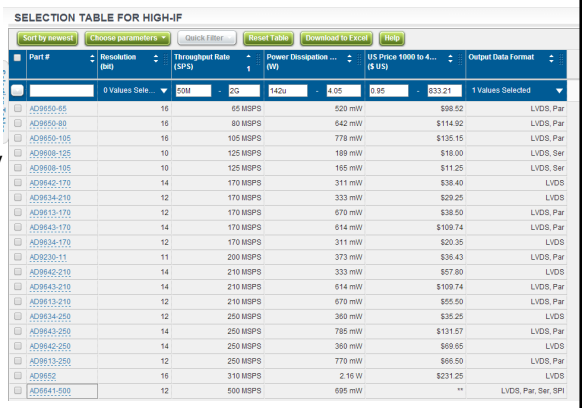
Data interfaces—conventional CMOS interfaces can't support data rates of much greater than 100 MHz—and low voltages differential swing (LVDS) data interfaces run up to 800 MHz to 1 GHz. For larger data rates, one can shift to multiple bus interfaces, or move to SERDES interfaces. Contemporary data converters are using SERDES interfaces at up to 12.5 GSPS (as specified in the JESD204B standard)—multiple data lanes can be used to support different combinations of resolution and speed in the converter interface. These interfaces can be quite sophisticated in their own right.

97

JEDEC Standard: JESD204 Serial Interface for ADC

ADC front-end in modern acquisition systems



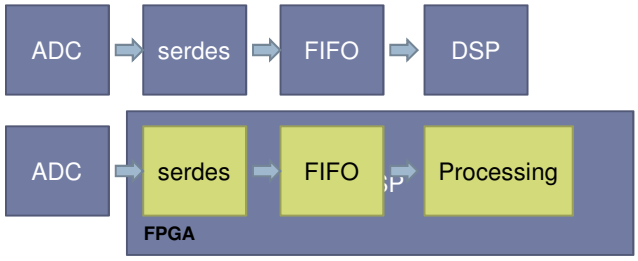


Dedicated modules are required for High Speed deserialization at the receiver
- DSP
-FPGA/ASIC

98

Data deserialization

•Adhoc Interfaces: Need to carefully read and understand the signaling and develop deserializing module
•Standard Interfaces: IP modules available for JESD204



99

Takeaways

- Rich set of families to address all areas of the FPGA market. In the xilinx case
  - Artix-7 family: Lowe price and power
  - Kintex-7 family: price/performance
  - Virtex-7 family: Highest performance/capacity
  - Zynq: ARM inside
- Unified architecture reduces learning curve for new designs
- Builds on the strengths of previous families
- Strong focus on power reduction
- Architectural features for the high DSP performance and low power

100

Further lectures

FPGA and CPLD architectures: a tutorial *Brown, S., Rose, J.*
IEEE Design & Test of Computers13(2)1996 pp:42-57

The DABC Framework Interface to Readout Hardware *J. Adamczewski-Musch, et al*
IEEE Transactions on nuclear science 58(4) 2011

Hardware-Based TCP Processor for Gigabit Ethernet, Tomohisa Uchida
IEEE Transactions on nuclear science (55): 3,

Highly scalable asynchronous trigger system for PET using UDP/IP K.E. Hines, et al
IEEE Nuclear Science Symposium Conference Record 2011 NP2.S-173

101

FPGA

SoC design for
Real time Signal Processing and High-Degree on-device data processing in Medical Imaging

Lessons so far

- FPGAs are mainly used
  - To compute point and neighborhood operations, because of their pipelined logic and on-board memory.
  - To implement reactive signaling with very constrained time requirements.
- DSPs/uP are mainly used
  - Control intensive (recursive operations) processing
  - Algorithm is subject to changes (programmability)
  - As the level of abstraction grows
- GPUs (discussed in other talks)
  - Are an alternative halfway

2nd International Summer School on:
INtelligent Signal Processing for FrontIEr Research and Industry

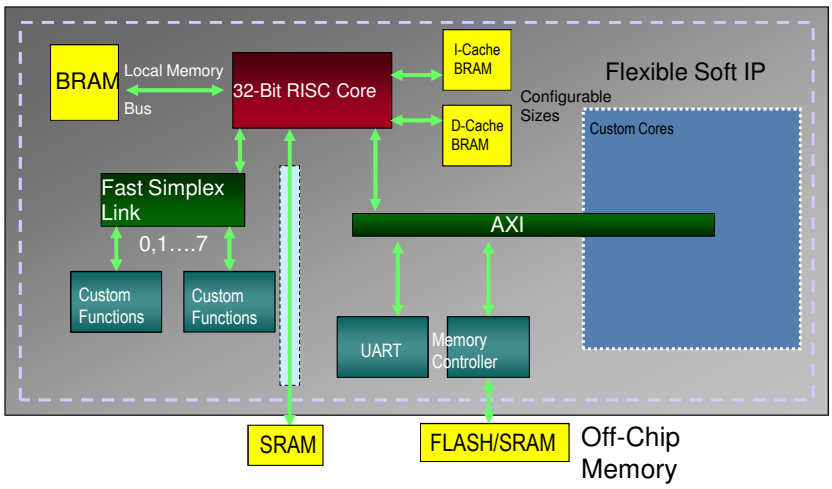
103

Embedded Development Enviroments

- Growing the level of abstraction: need for fast development -> IP based
- Major FPGA vendors provide design environments to create embedded SoC designs
  - FW: IP based (with VHDL behing)
    - Processor sub-system component of larger design
  - SW: SDK software design environment (Eclipse-based)
    - Enables the integration of both hardware and software components of an embedded system
- This approach enables an integrated solution for real-time processing
  - Dedicated HW for data intensive algorithms
  - SW for control intensive algorithms with co-processors for acceleration

Page

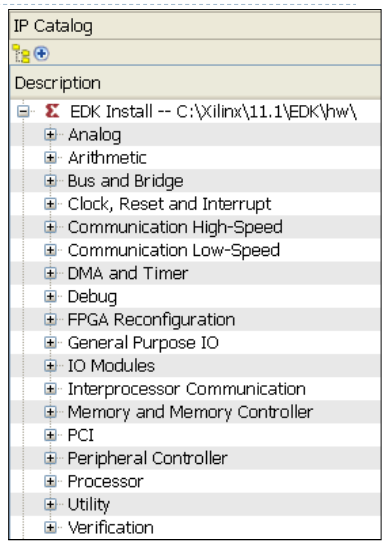
Processor-Based Embedded Design



Page

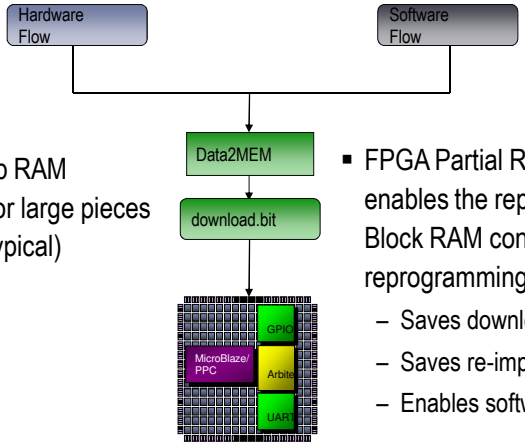
IP Peripherals

- Bus infrastructure and bridge cores
- Memory and memory controller cores
- Debug
- GPIO Peripherals
- Arithmetic
- Timers
- Inter-processor communication
- External peripheral controller
- DMA controller
- PCI
- Video/Image Processing
- User define core



Page 106

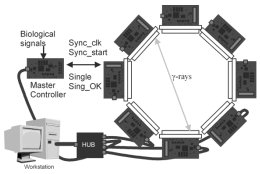
Merging Hardware and Software Flows



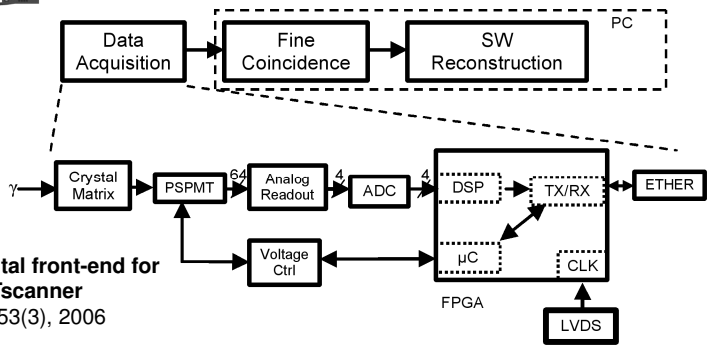
- Use of off-chip RAM encouraged for large pieces of software (typical)
- FPGA Partial Reconfiguration enables the reprogramming of the Block RAM contents without reprogramming the entire FPGA
  - Saves download time
  - Saves re-implementation time
  - Enables software debugging

Page

Application Example: PET ACQ Front-end



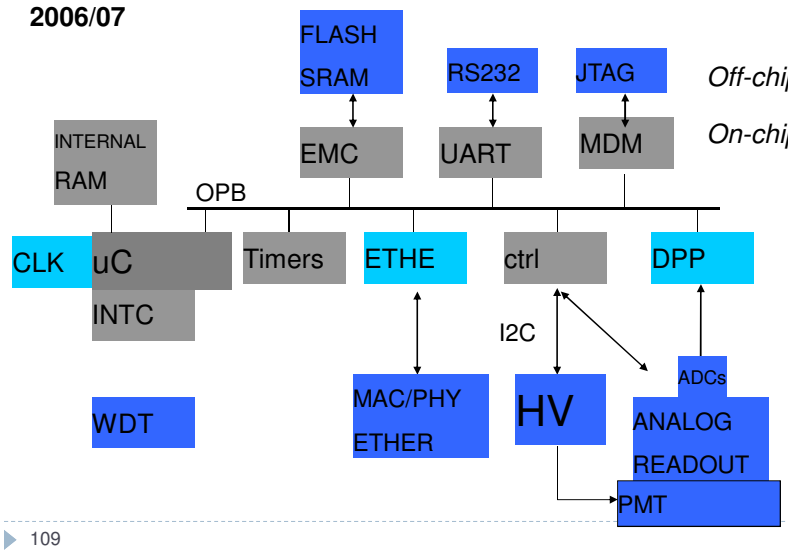
Smart detector for PET



New embedded digital front-end for high resolution PETscanner
Guerra, P. et al TNS 53(3), 2006

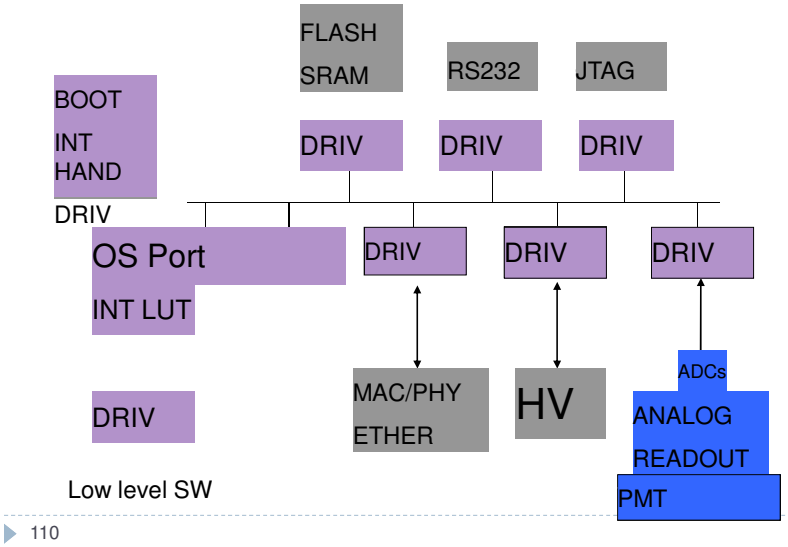
108

SoC: HW Architecture



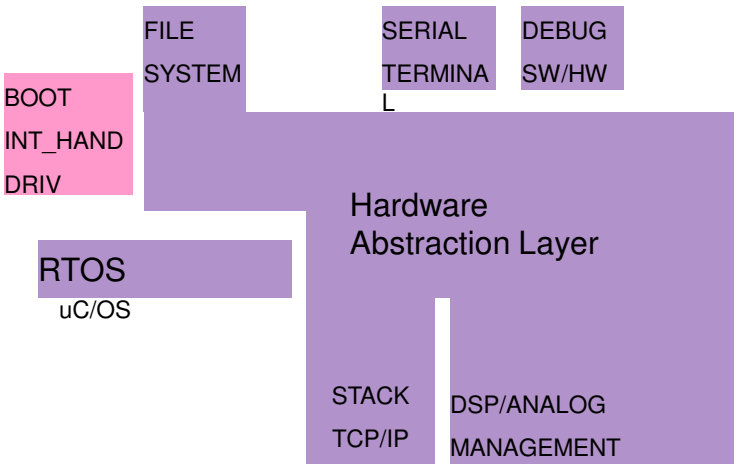
109

SoC Architecture: Low Level SW - Abstraction



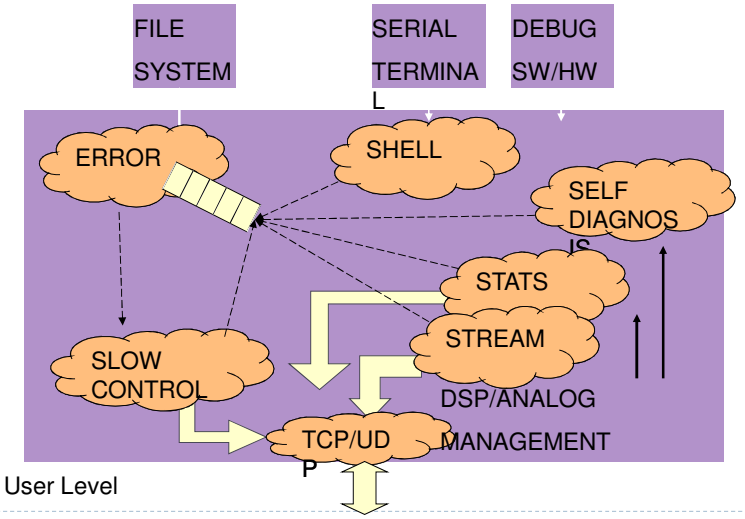
110

SoC Architecture: HAL SW



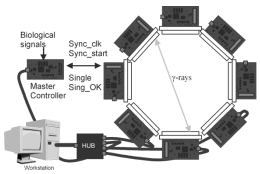
111

SoC Architecture: Task Level SW

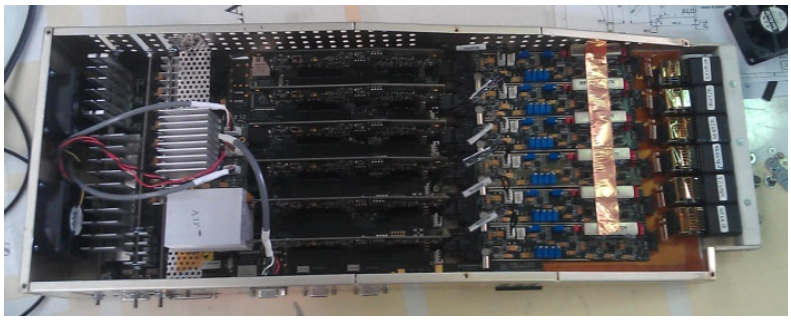


112

Application Example: PET ACQ Front-end



2014:
No RTOS
NO generic TCP stack
uC very useful for slow control



113

Day Takeaways

- **Real time Signal Processing**
  - Floating to fixed point algorithm transformation
  - Parallelism is the key: architecture
    - DSP: multiple functional units (VLIW) : compiler & programming
    - FPGA/ASIC: dedicated architecture design : need to understand your device and constraints.
- **High-Degree on-device data processing**
  - Complex algorithms can be integrated on chip
  - IP reusing becomes a must-do: Focus on the core
  - uP provide flexibility to select the most efficient approach to the implementation.

114

That's all folks