CE/CZ 3001:
Advanced Computer Architecture

(Module 8: Custom Computing and Emerging Computing Trends)

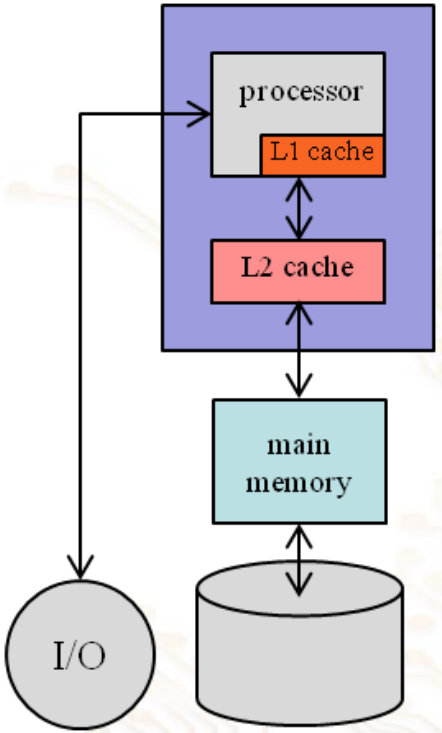
Dr Smitha K. G.
School of Computer Science
and Engineering

Overview

- Applications of specific architectures
  - Examples: ASIP, FPGA and ASIC
  - Comparison of general purpose processors, DSP, GPU, FPGA and ASIC
- Heterogeneous multicore platforms
  - Introduction to domain specific computing

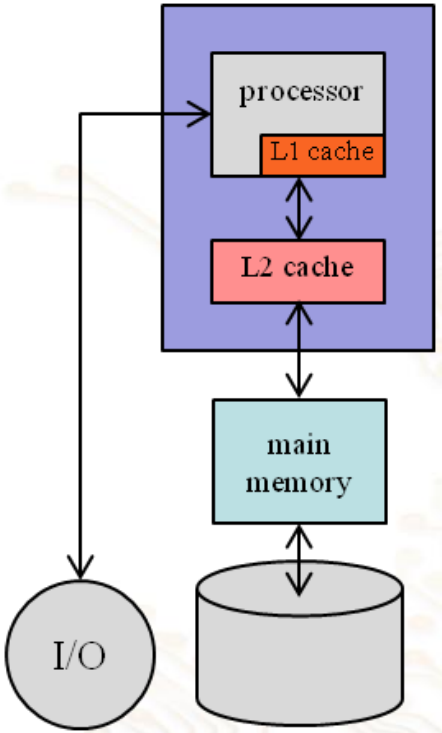
General Purpose Processor (GPP) (Part 1/2)

- Hardware features
  - The program to be run and necessary data could be made available at the main memory
  - General data-path: consists of a general ALU and usually a large register file
  - Multiple levels of cache for reducing memory latency



General Purpose Processor (GPP) (Part 2/2)

- Maximum flexibility
  - Programmable: supports several high-level languages
  - Can be used for any general application
- Other key features
  - Short time-to-market
  - Low NRE cost
  - High power consumption



Application Specific Instruction Set Processor (ASIP) (Part 1/2)

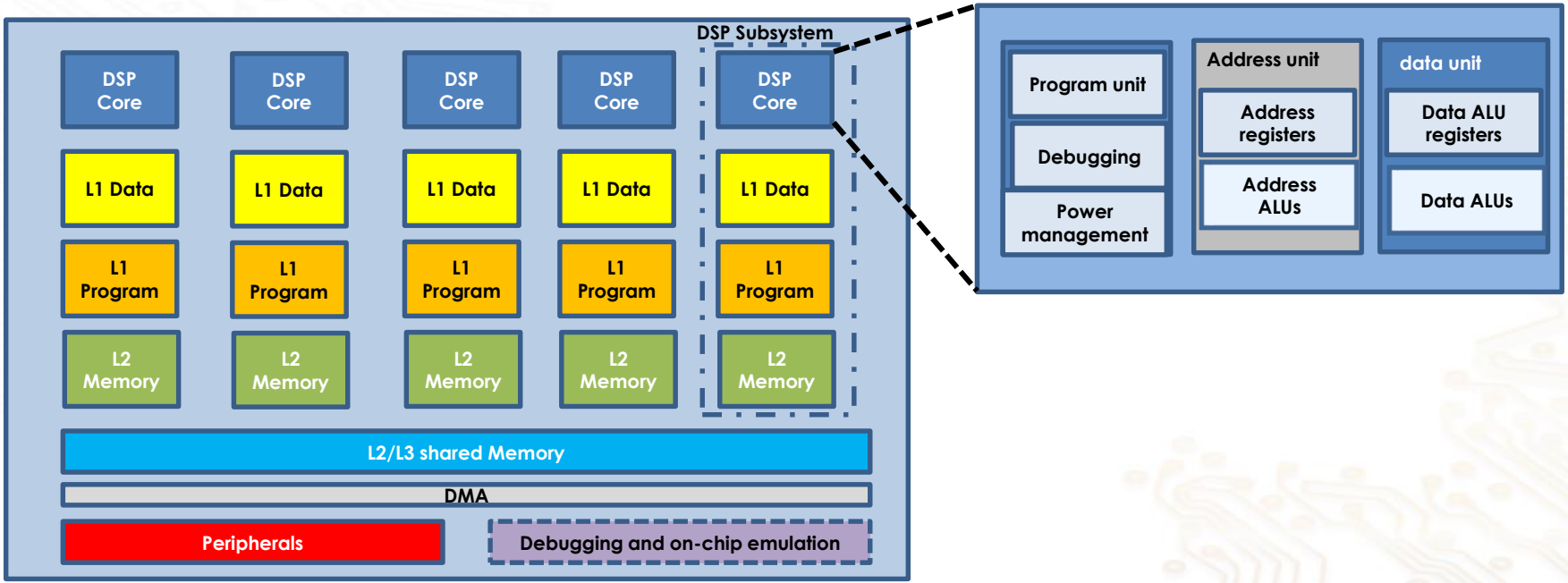
- Application specific instruction set processor (ASIP): a microprocessor tailored to benefit a specific application or a domain of applications
  - Signal processing, image processing, video processing, digital communication, etc.
- Instruction set of ASIP is customised for the type of computation involved in the specific application

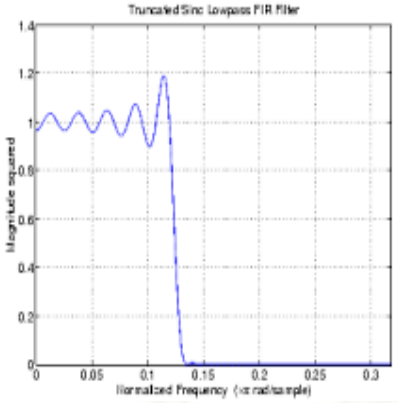
Application Specific Instruction Set Processor (ASIP) (Part 2/2)

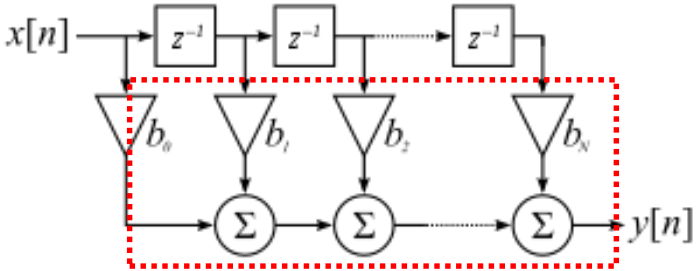
- Requirement on flexibility should be just sufficient instead of unlimited like general purpose processor
- To achieve highest performance with minimum power consumption, silicon cost and design cost

Digital Signal Processor (DSP) (Part 1/3)

A Digital Signal Processor (DSP) is an ASIP designed for repetitive multiply-accumulate operation and bit-reversal addressing.







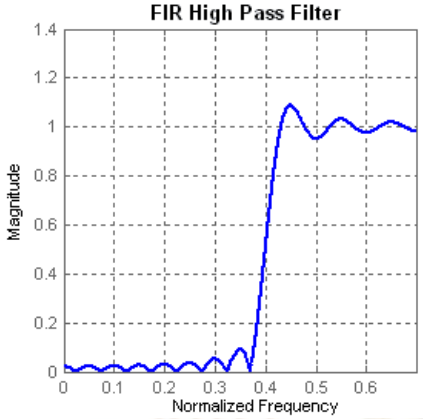


Photo of person retrieved from https://pxhere.com/en/photo/367883 Used under CC0 (https://creativecommons.org/share-your-work/public-domain/cc0/)

Digital Signal Processor (DSP) (Part 2/3)

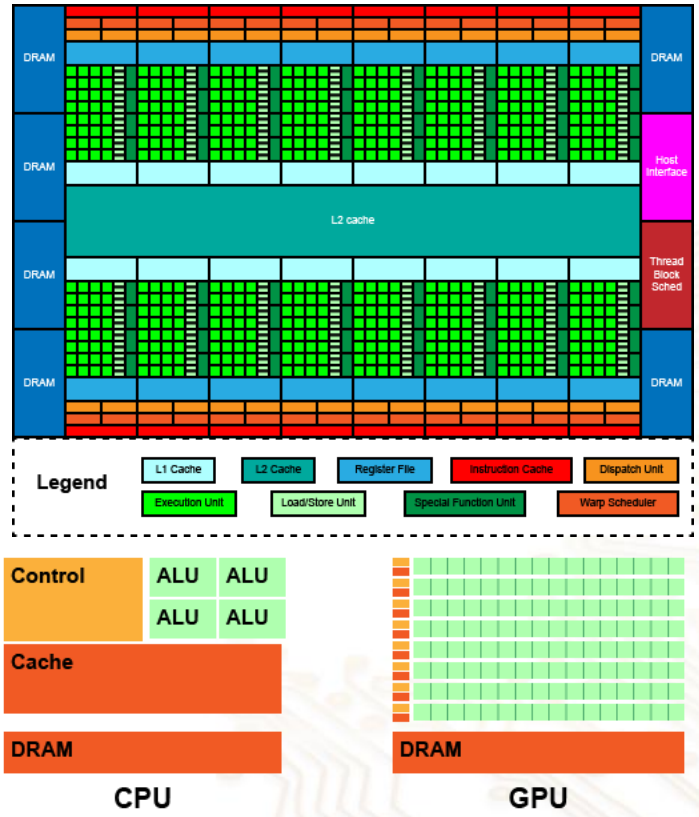
- Real time processing of computation-intensive tasks while minimising:
  - Cost: lower cost in comparison with general purpose processors (GPP)
  - Power consumption
  - Specialised data-path: e.g. multiply-accumulate (MAC) unit
  - Specialised instruction set for digital signal processing

Digital Signal Processor (DSP) (Part 3/3)

- Repetitive numeric calculations
  - Specialised execution control
  - Fixed-point implementation with good numeric fidelity
- High memory bandwidth
- Supports streaming data
  - Special mechanism for real time I/O
- Less development time

Graphic Processing Unit (GPU) (Part 1/3)

- GPU is a processor — specialised for graphics
- Modern GPUs are very efficient at manipulating computer graphics and image processing, and their highly parallel structure makes them more efficient than general-purpose CPUs for algorithms where the processing of large blocks of data is done in parallel
- Heterogeneous execution model (CPU is the host, GPU is the device)



Graphic Processing Unit (GPU) (Part 2/3)

- The motivation for GPU computing
  - Parallel computing by massively data parallel stream processing [1]
  - Using thousands of threads
  - Implemented in hundreds of cores
  - Cost effective: cheap
  - Using already available hardware

[1] http://en.wikipedia.org/wiki/Stream_processing

Graphic Processing Unit (GPU) (Part 3/3)

- GPUs involve high latency and provide high throughput processing
- CPU performs low latency low throughput computation
- Embedded systems are usually real time

Field Programmable Gate Array (FPGA)

Field Programmable Gate Array (FPGA) (Part 1/2)

- The core architecture of FPGA consists of three main components:
  - An array of configurable/ programmable logic blocks: the combinational units
  - A sea of programmable interconnects
  - Memories and specialised I/O blocks

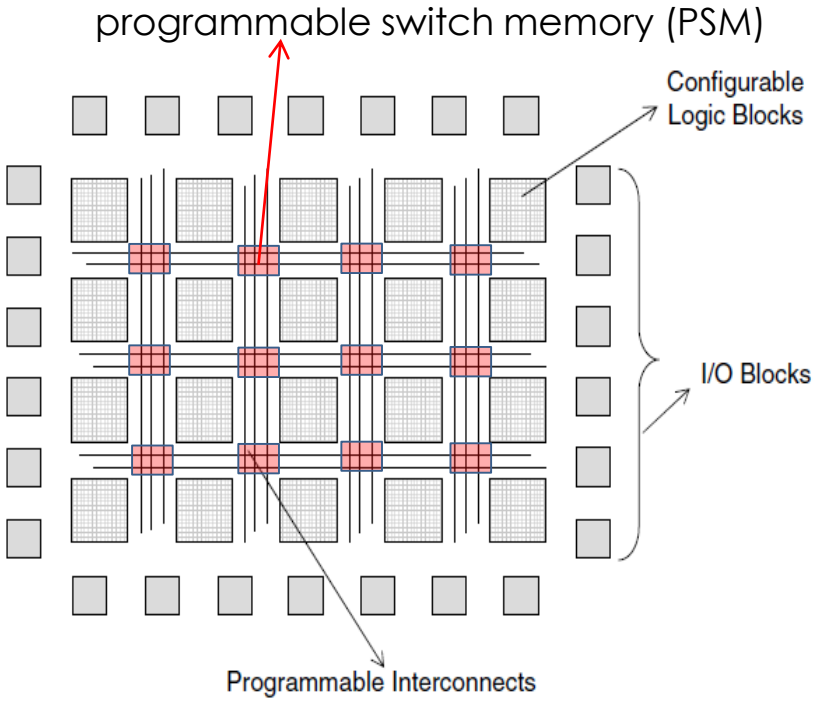


Figure 2. FPGA structure. Adapted from "FPGA and CPLD Architectures: A Tutorial" (p. 44) by S. Brown and J. Rose, 1996, IEEE Design & Test of Computers, Summer 1996, pp. 42-57.

Field Programmable Gate Array (FPGA) (Part 2/2)

- Other components in modern FPGAs are:
  - DSP blocks
  - Hardwired IP cores
  - Soft cores

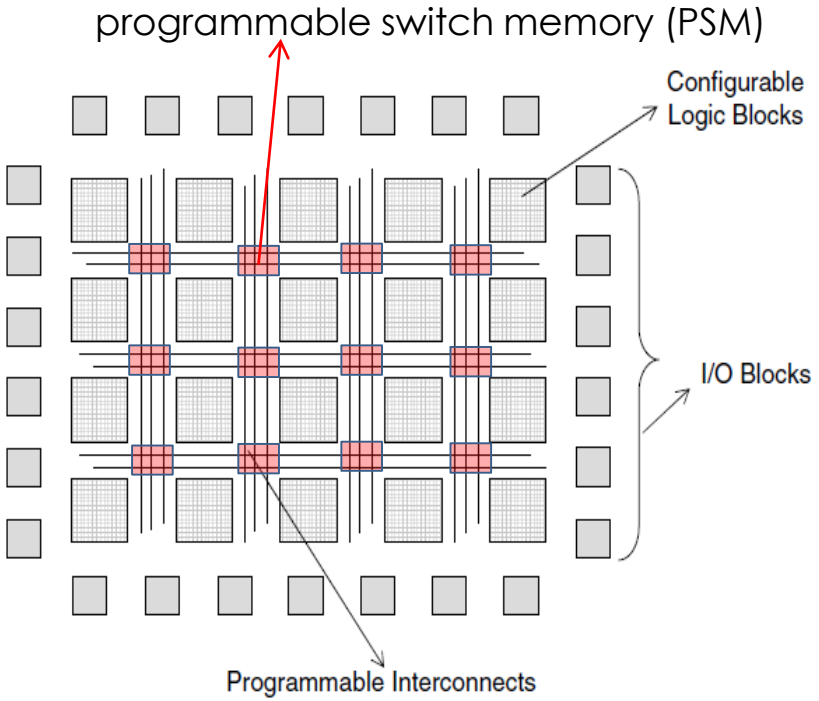


Figure 2. FPGA structure. Adapted from "FPGA and CPLD Architectures: A Tutorial" (p. 44) by S. Brown and J. Rose, 1996, IEEE Design & Test of Computers, Summer 1996, pp. 42-57.

Field Programmable Gate Array (FPGA) (Part 1/2)

- Configurable Logic Block (CLB): Contains two 4-input LUTs and a third LUT (3-input) fed by the output of other two.
  - CLB can implement logic functions
  - Functions with more than four inputs or two separate 4-input functions
  - Each LUT is 1-bit-wide memory array

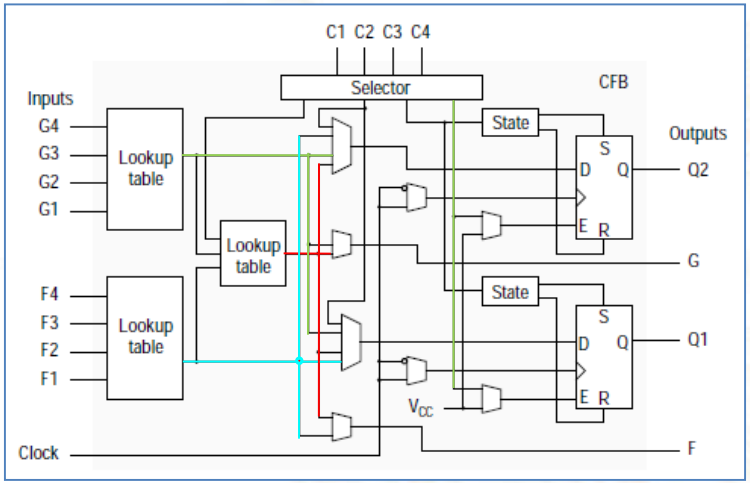


Figure 18. Xilinx XC 4000 CLB. Adapted from "FPGA and CPLD Architectures: A Tutorial" (p. 52) by S. Brown and J. Rose, 1996, IEEE Design & Test of Computers, Summer 1996, pp. 42-57.

Field Programmable Gate Array (FPGA) (Part 2/2)

- Configurable Logic Block (CLB): Contains two 4-input LUTs and a third LUT (3-input) fed by the output of other two.
  - Address lines are CLB inputs
  - Any desired K-input logic function can be realised this LUT by writing the truth table for the function directly into the memory

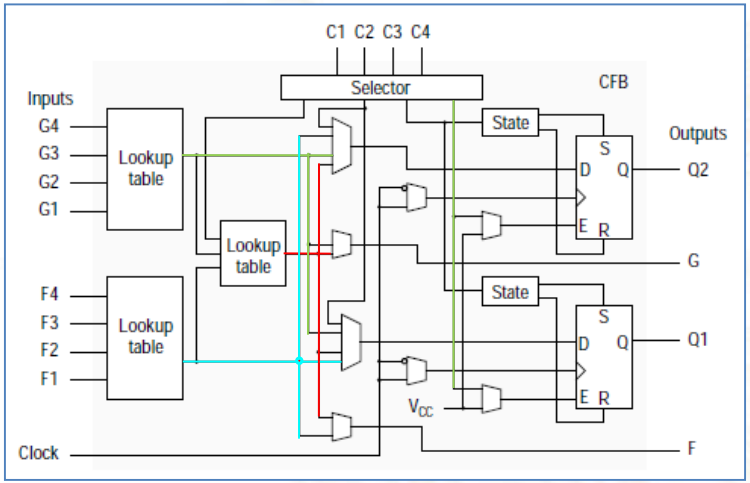
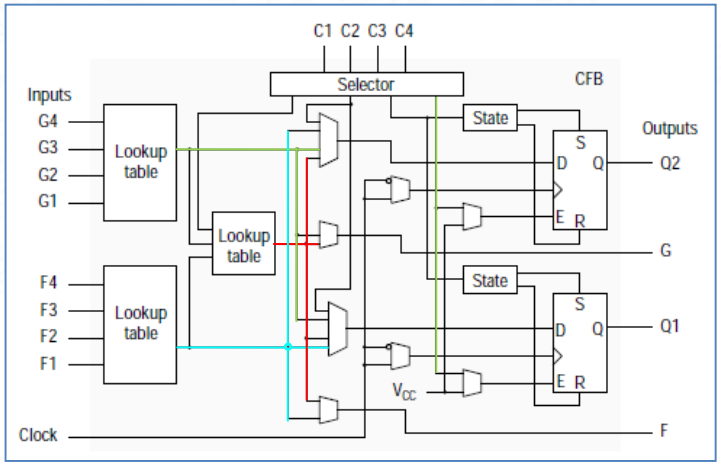


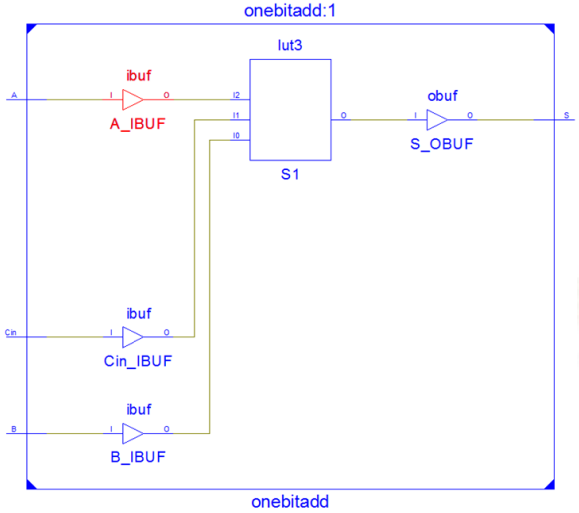
Figure 18. Xilinx XC 4000 CLB. Adapted from "FPGA and CPLD Architectures: A Tutorial" (p. 52) by S. Brown and J. Rose, 1996, IEEE Design & Test of Computers, Summer 1996, pp. 42-57.

FPGA — How it works (Part 1/3)

For a single bit full adder
S=A (XOR) B (XOR) Cin

Number of inputs = 3
Number of outputs = 1





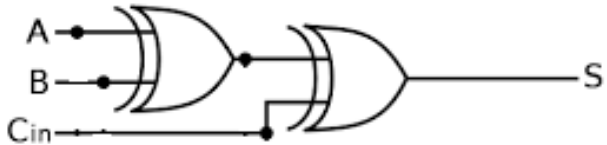
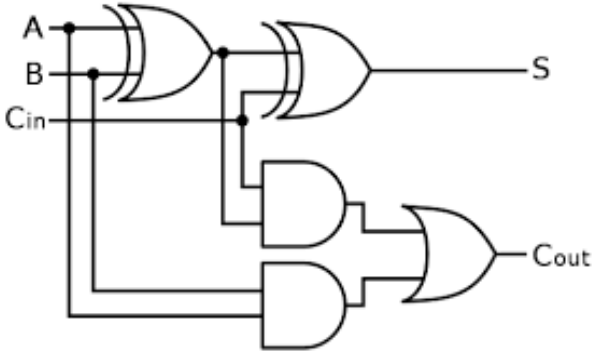


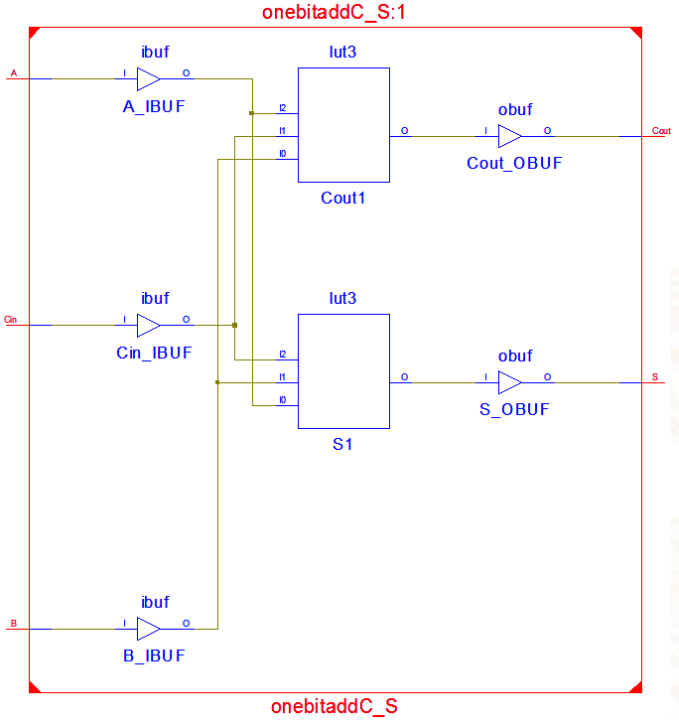
Figure 18. Xilinx XC 4000 CLB. Adapted from "FPGA and CPLD Architectures: A Tutorial" (p. 52) by S. Brown and J. Rose, 1996, IEEE Design & Test of Computers, Summer 1996, pp. 42-57.

FPGA – How it works (Part 2/3)

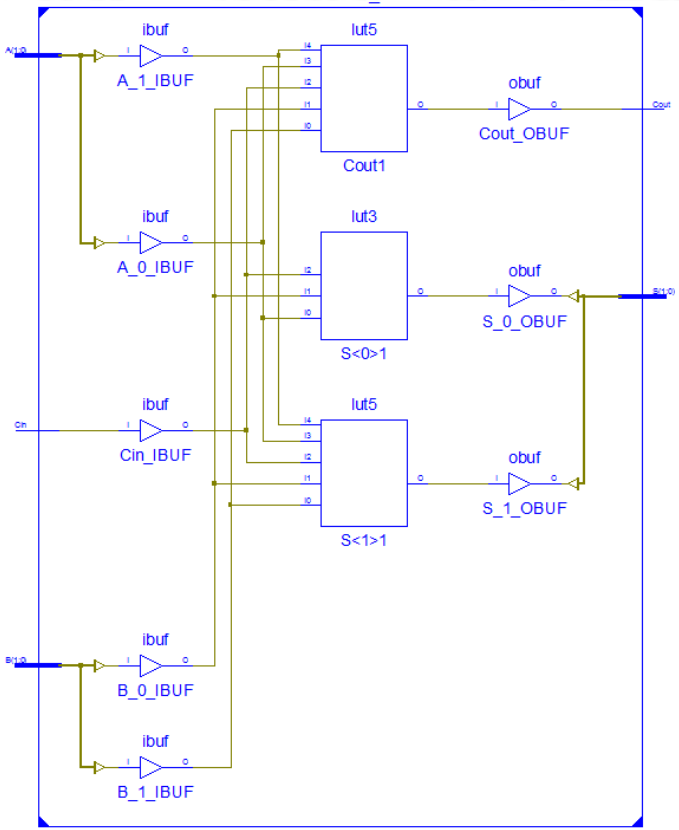


Sum=A (XOR) B (XOR) Cin
Cout= (A (XOR) B) Cin+ AB

Number of inputs = 3
Number of outputs = 2



FPGA – How it works (Part 3/3)



Number of inputs = 5
Number of outputs = 3

No of LUTs = 3

FPGA vs DSP (Part 1/2)

- Both FPGA and DSP offer flexibility for reuse and programmability
- DSP has an advantage in time-to-market
- FPGA supports parallel design and offers greater performance compared to DSP
- FPGA is more expensive than DSP

FPGA vs DSP (Part 2/2)

- Power consumption of FPGA is higher than DSP
- Development time for FPGA is longer and more complicated than DSP
- Heterogeneous architectures, consisting of both DSP and FPGA components is an emerging trend

Application Specific Integrated Circuits (ASIC)

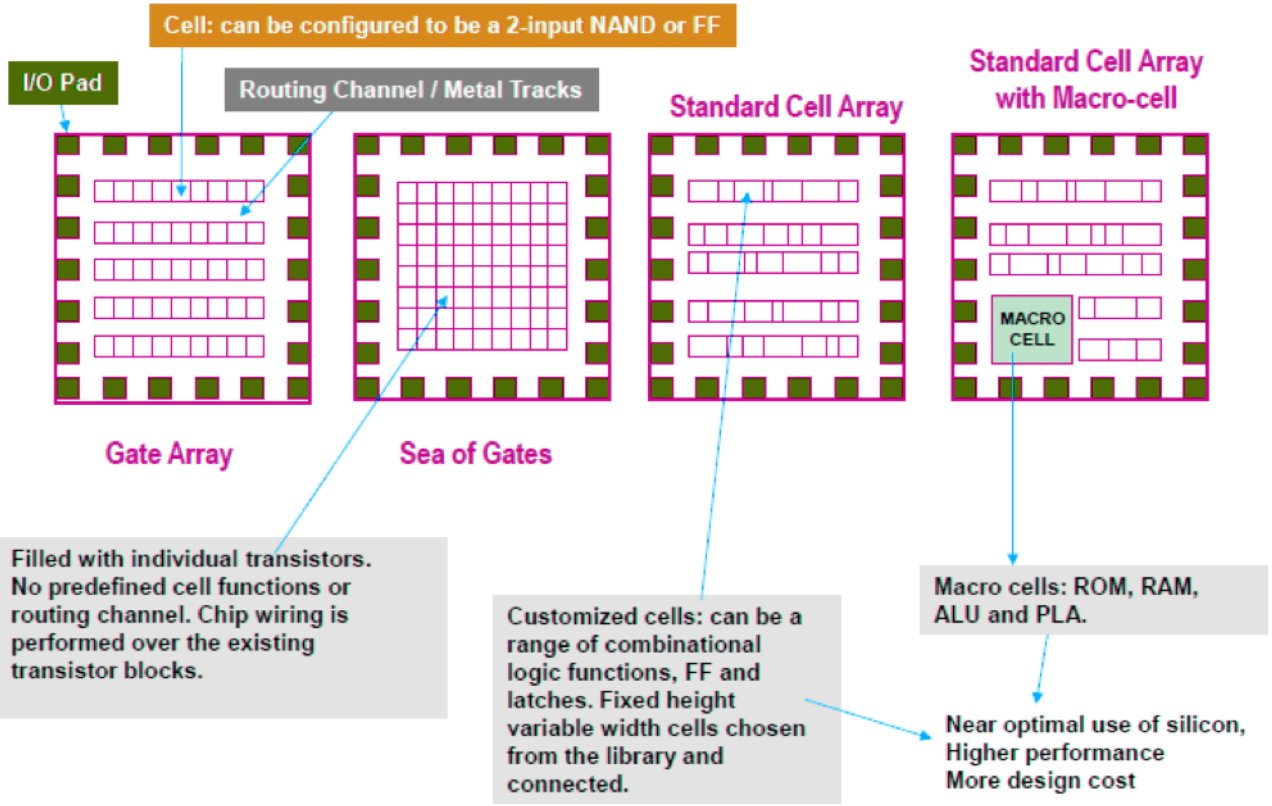
Application Specific Integrated Circuits (ASIC) (Part 1/2)

- Provides customised tailored hardware solutions for specific applications or problems
  - Not configurable: inflexible
  - Cannot be upgraded, updates requires redesign
  - Involves high NRE cost of several millions for development and testing (need to have large volume market to reduce cost)
  - Longer design and development time: longer time-to-market
  - Maximum performance per watt, per unit area

Application Specific Integrated Circuits (ASIC) (Part 2/2)

- Provides customised tailored hardware solutions for specific applications or problems
  - Mixed-analog and digital design
  - Design optimisation is possible
- Semi-custom design: gate array implementation: standard cell implementation
- Full-custom design: cell design, cell library development and use: simulation and testing for design verification

Gate arrays and standard cells



FPGA vs ASIC (Part 1/2)

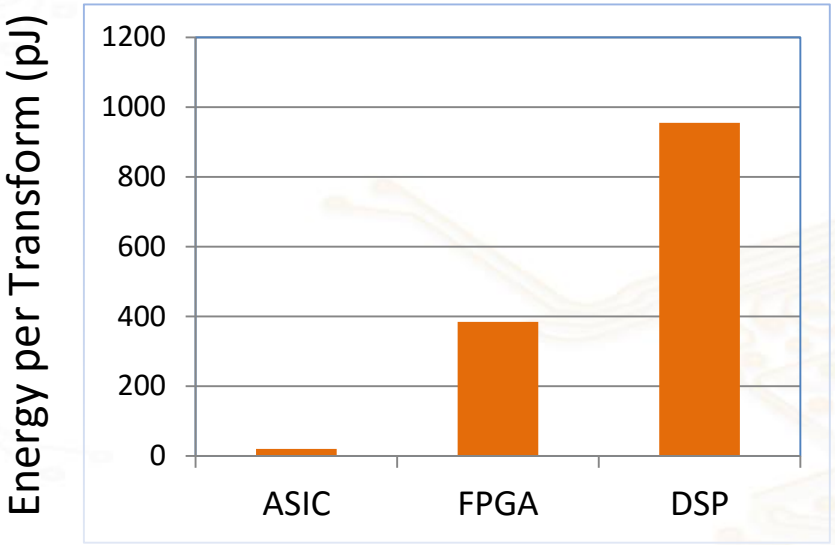
- FPGA and ASIC: highly parallel tasks to provide greater performance
- ASIC can provide higher clock period, thus higher speed than FPGA
- ASIC has a smaller form factor than FPGA
- FPGA consumes more power and energy than ASIC

FPGA vs ASIC (Part 2/2)

- FPGA is more expensive than ASIC for large volume market
- FPGA is reconfigurable (design reuse, mistakes rectifiable) while ASIC is not flexible at all
- FPGA has shorter development time (faster time-to-market)
- ASIC development tools are expensive

Comparison of energy consumption

- Energy and area efficiency achieved by direct mapping of algorithm to ASIC is as high as 1000 times of programmable processors
- Cost of an ASIC design in 45-nm CMOS technology is more than $50,000,000
- Design cycle for ASIC can easily exceed a year



Energy consumption for 64-point FFT using 0.18µ ASIC, FPGA and a high-performance DSP.

Comparison of processor technologies

Performance/unit power consumption vs flexibility

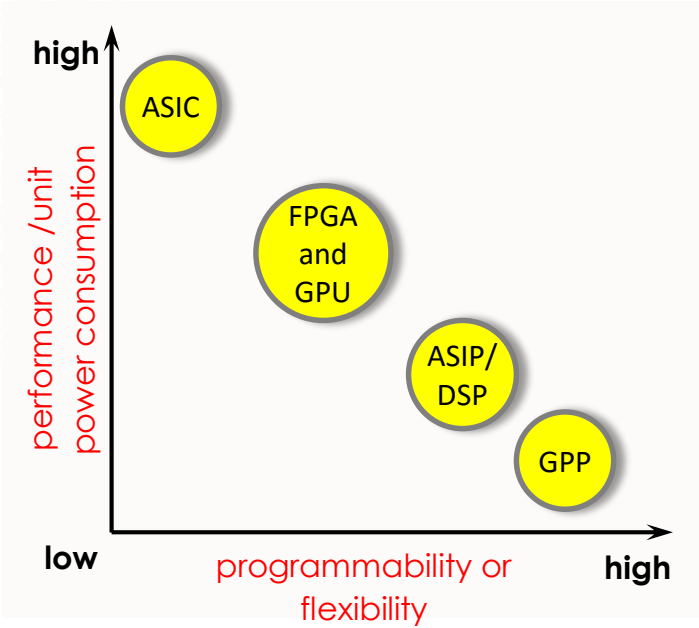


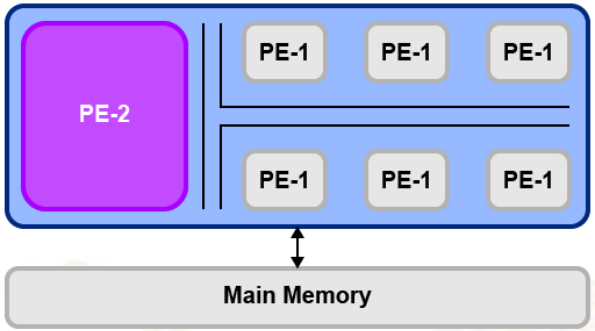
Figure 6.1. A simplified comparison of different technologies: programmability versus performance. Adapted from Computer System Design: System-on-Chip (p. 209), by M. J. Flynn & W. Luk, 2011, Hoboken, N.J.: Wiley.

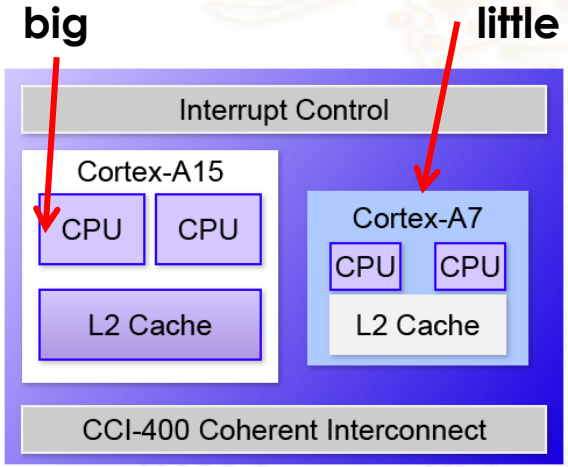
Heterogeneous Multicore Platforms

Heterogeneous computing systems: ARM big.LITTLE

Example of multicore system
with two different types of cores:

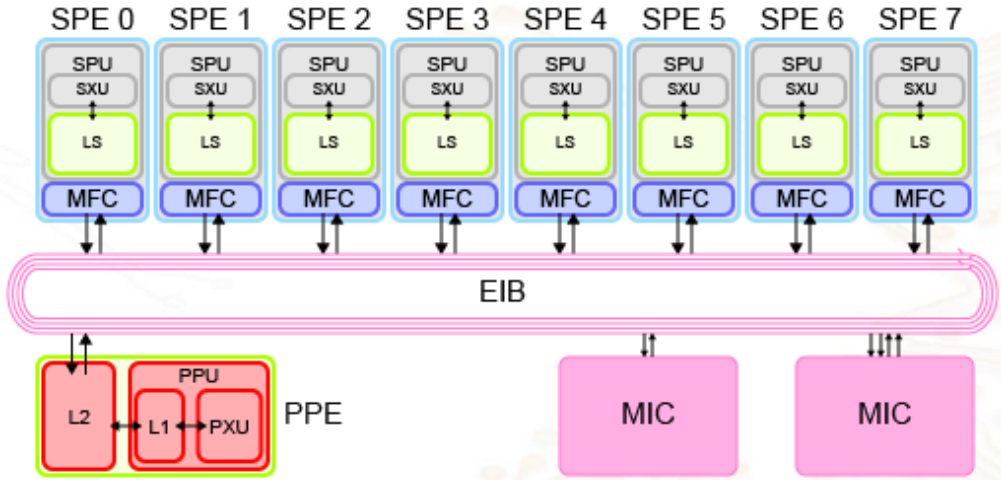
- Selects the processor according to the job to minimise power consumption
- **Little:** ARM Cortex-A7: consists of in-order, 8-stage pipeline CPUs
  - Always on, performs the tasks which demand less computation
- **Big:** ARM Cortex-A15: consists of out-of-order multi-issue pipeline high-performance CPUs
  - Does the computation-intensive tasks
  - Need not be always on





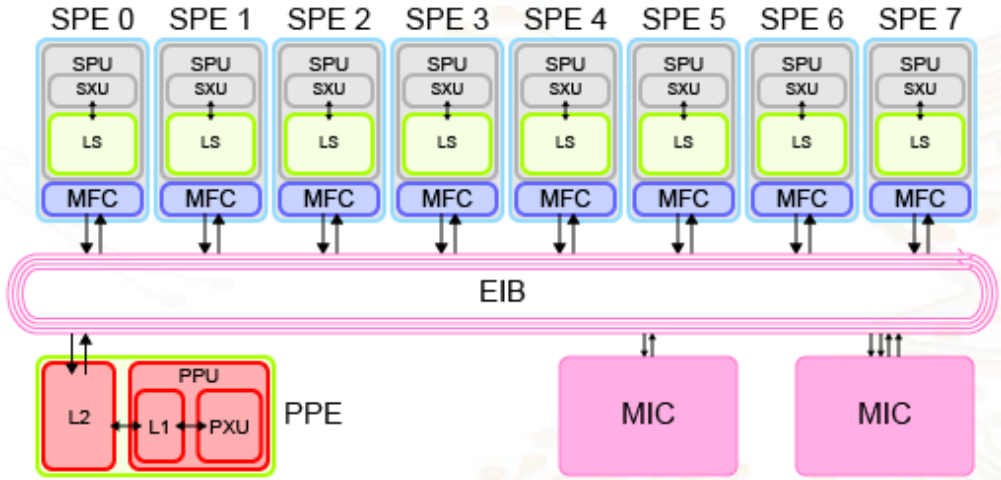
Heterogeneous computing systems: IBM Cell Broadband Engine (BE) architecture (Part 1/4)

- A heterogeneous multicore system with two types of cores: customised for gaming/graphics rendering.
- Sony-Toshiba-IBM (STI) partnership built the CELL processor that was used in Sony's PlayStation 3 (2000-2005).



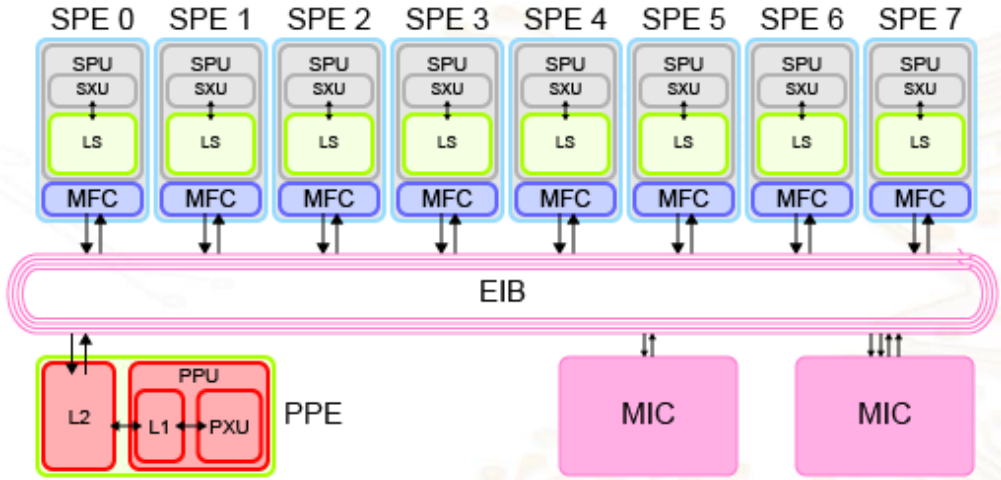
Heterogeneous computing systems: IBM Cell Broadband Engine (BE) architecture (Part 2/4)

- The Power Processor Element (PPE) assigns tasks to the other cores and controls the communication between the cores.



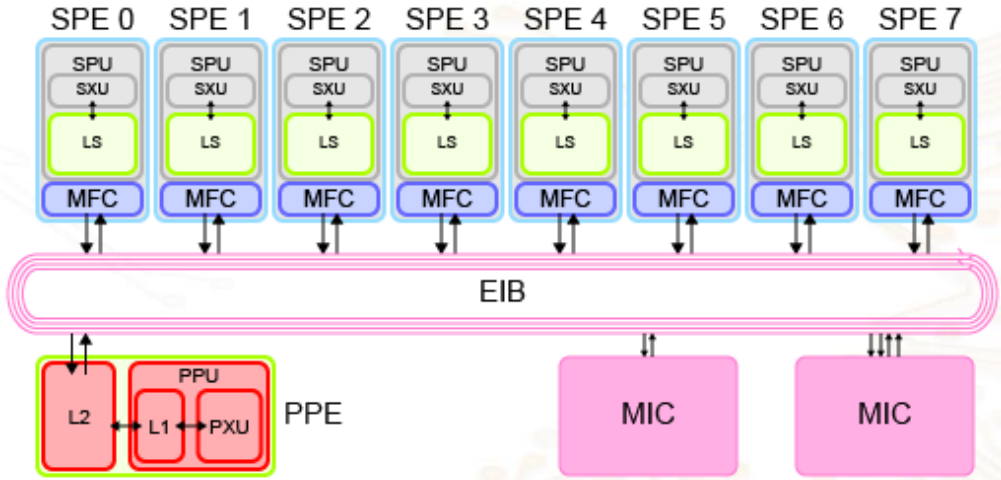
Heterogeneous computing systems: IBM Cell Broadband Engine (BE) architecture (Part 3/4)

- Synergistic Processor Elements (SPE) perform vector operations: designed to provide high floating-point throughput.
- Eight independent SPEs, each having 256KB of local storage and using128-bit SIMD instructions.



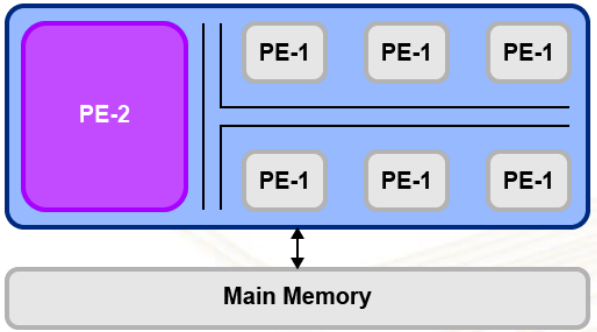
Heterogeneous computing systems: IBM Cell Broadband Engine (BE) architecture (Part 4/4)

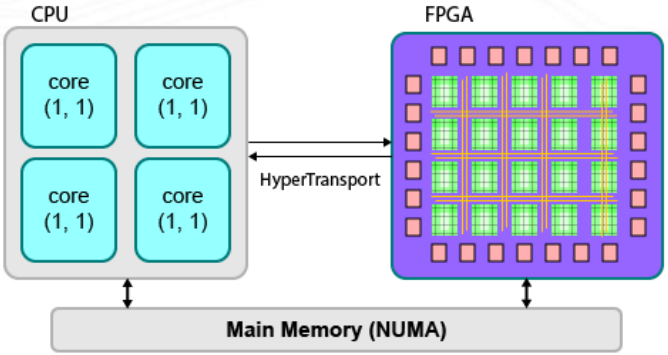
- Synergistic processor unit (SPU)
- Element interconnect bus (EIB) (up to 96 bytes/cycle)
- Memory flow controller (MFC)
- Memory interface controller (MIC)
- Bus interface controller (BIC)

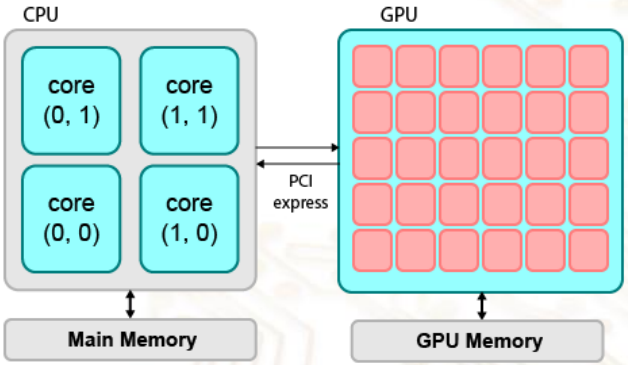


General heterogeneous computing systems

Heterogeneous systems can combine different processing technologies such as FPGA, ASIC, general-purpose processors, graphics processing units, digital signal processors or microcontroller units to achieve desired performance with less power consumption.







Domain Specific Computing

Domain Specific Computing (DSC)

- A domain specific computing system is a heterogeneous multicore system customised specifically according to the computations (the kind of processing) involved in a given application domain for high-performance and power efficient realisation of the computations for the particular application domain and support the changes in the application domain.

- A domain specific computing system may contain some generic programmable core, customised cores, hardware accelerators and programmable interconnects.

Motivation for Domain Specific Computing (Part 1/2)

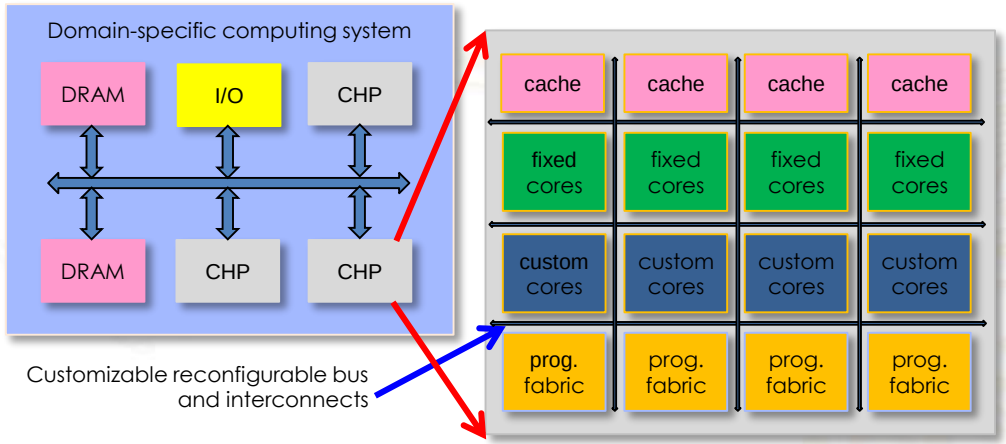
- Each user has a high computing demand in one or a few selected application domains
  - For example: **graphics** for game developers, **circuit simulation** for integrated circuit design houses, **financial analytics** for investment banks, **transforms and filters** for signal/image and video processing, **matrix operations** for weather forecasting, scientific computing, fluid dynamics
  - General computing needs like email, word processing, web browsing can be easily achieved by any existing computing technology

Motivation for Domain Specific Computing (Part 2/2)

- To develop customisable computing systems for a particular application domain that gives significant improvements in power-performance efficiency
- To make the best use of other technologies like ASIC, FPGA or ASIP along with the programmable processors to achieve the best power-performance efficiency with lower cost

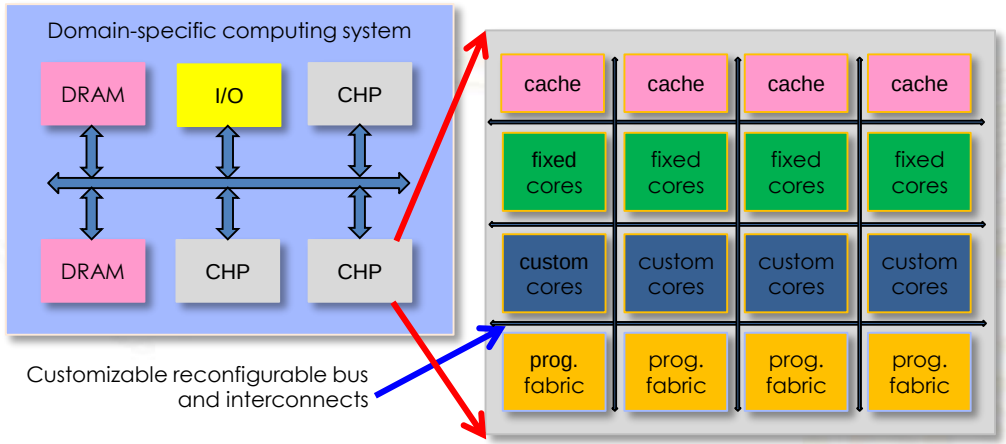
Example of a Domain Specific Computing (DSC) system (Part 1/2)

- Low-latency, high-bandwidth interconnects for data sharing between cores, co-processors, cache banks and memory banks to meet the communication requirements of a particular application (or even different phases of the same application).



Example of a Domain Specific Computing (DSC) system (Part 2/2)

- CHP is characterised by three types of components (fixed cores, customisable cores, and programmable fabrics), each having different levels of customisation, re-configurability and parallelism.



Customisable heterogeneous platform of a DSC system (Part 1/2)

- Fixed cores: differ widely in terms of their energy efficiency, performance and area
  - They have very limited (or no) re-configurability
  - Allow voltage/frequency scaling to adapt power/performance
- Customisable cores: architecture customised (tuned) according to the application
  - Offer a set of tunable options such as register file sizes, cache sizes, bit-width of datapath operating frequency and supply voltages

Customisable Heterogeneous Platform of a DSC System (Part 2/2)

- Programmable fabrics: provide maximum flexibility
  - Implement custom instructions by specialised co-processor (FFT, DCT, filters, etc.) to accelerate performance
  - Architecture customised in terms of number of computing units, types of computing units, number of pipeline stages, etc. to implement complex operations

Processor technology summary (Part 1/2)

- Programmable processors
  - General purpose processor
    - Single-core and multicore processors
  - Specialised programmable processor
    - Application Specific Instruction Set processor (ASIP), GPU, DSP
- Dedicated processors
  - Reconfigurable processor (e.g. FPGA)
  - Application specific integrated circuits (ASIC)

Processor technology summary (Part 2/2)

- Choice of implementation platforms
  - System cost and volume of production
  - Speed performance, power consumption
  - Reuse of design, reuse of system or subsystems (flexibility)
  - Time-to-market vs design and development time
- Heterogeneous systems on chip (SoC)