Reconfigurable Computing

Tran Ngoc Thinh
HCMC University of Technology
http://www.cse.hcmut.edu.vn/~tnthinh/rc

Administrative Issues

- Class
  - Time and venue: Wednesdays, 6:30am - 09:00am, 303B4
  - Web page:
    - http://www.cse.hcmut.edu.vn/~tnthinh/rc
  - Textbook:
    - Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications. Tác giả: Christophe Bobda. 2007
    - Reconfigurable Computing: Accelerating Computation with Field-Programmable Gate Arrays. Tác giả: Maya Gokhale and Paul S. Graham. 2005
    - Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing, Tác giả: Scott Hauck and André Dehon. 2008
    - Một số bài báo của các tạp chí chuyên ngành IEEE, ACM Transactions, và các hội nghị quốc tế về tính toán tái cấu hình FCCM, FPL, FPT, FPGA…

Administrative Issues (cont.)

- Grades
  - 30% presentations
  - 30% mini project
  - 40% final exam

Administrative Issues (cont.)

- Personnel
  - Instructor: Dr. Tran Ngoc Thinh
    - Email: tnthinh@cse.hcmut.edu.vn
    - Phone: 8647256 (5843)
    - Office: A3 building
    - Office hours: Tuesdays, 08:30-10:00
  - TA:
    - Email:
    - Phone: _____
    - Office: _____
    - Office hours: _____

Chapter 0
Introduction, Motivation, Goals

Introduction

☐ Computer anytime anywhere (pervasive and ubiquity) …

Introduction

- ... communication also.

Introduction

- Explosive growth in
  - Computing
  - Communication
- Information technology
  - „Hand in hand com" growth in computing and communication
- Millions of computer systems are produce every year
  - PCs, Laptops, Workstations, Mainframes, Server
- Billion of embedded computer systems already deployed
  - Household, cars, mobile phone, plane, etc…

Computing Paradigms

- The Von Neumann Computer
- Domain specific processors
- Application specific processors
- Reconfigurable Processors

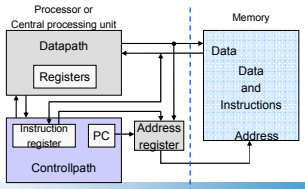
The Von Neumann Computer

- Principle

  In 1945, the mathematician Von Neumann (VN) demonstrated in study of computation that a computer could have a simple structure, capable of executing any kind of program, given a properly programmed control unit, without the need of hardware modification

The Von Neumann Computer

- Structure
  - A memory for storing program and data.
    - The memory consists of the word with the same length
  - A control unit (control path) featuring a program counter for controlling program execution
  - An arithmetic and logic unit (ALU) also called data path for program execution



The Von Neumann Computer

- Coding

  A program is coded as a set of instructions to be sequentially executed

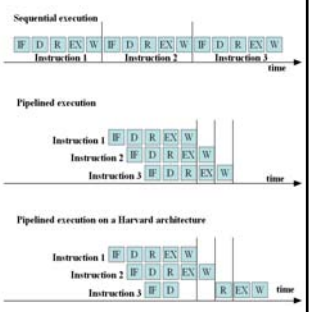
- Program execution
  - Instruction Fetch (IF): The next instruction to be executed is fetched from the memory
  - Decode (D): The instruction is decoded to determine the operation
  - Read operand (R): The operands are read from the memory
  - Execute (EX): The required operation is executed on the ALU
  - Write result (W): The result of the operation is written back to the memory
  - Instruction execution in Cycle (IF, D, R, EX, W)

The Von Neumann Computer

- Advantage:
  - Flexibility: any well coded program can be executed
- Drawbacks
  - Speed efficiency: Not efficient, due to the sequential program execution (temporal resource sharing).
    - Resource efficiency: Only one part of the hardware resources is required for the execution of an instruction. The rest remains idle.
    - Memory access: Memories are about 10 time slower than the processor
  - Drawbacks are compensated using high clock speed, pipelining, caches, instruction pre-fetching, etc.

The Von Neumann Computer

- Sequential execution
- t_{cycle} = cycle execution time
- One instruction needs $t_{instrcution} = 5*t_{cyc}$
- 3 instructions are executed in $15*t_{cycle}$
- Pipelining:
- One instruction needs $t_{instrcution} = 5*t_{cyc}$
- no improvement.
- 3 instructions need $7*t_{cycle}$ in the ide
- case.
- $9*t_{cycle}$ on a Harvard architecture.
- → Increased throughput



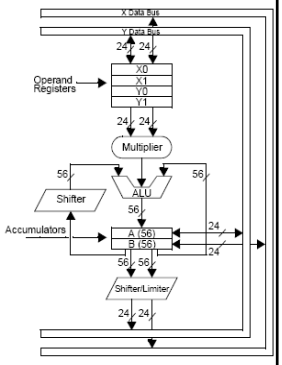
- Even with pipeline and other improvement like cache, the execution remain sequential.

Domain specific processors

- Goal: Overcome the drawback of the von Neumann computer.
- Optimized Datapath for a given class of applications
- Example: DSP (Digital Signal Processors): Signal processing applications are usually multiply accumulate (MAC) dominated.
  - Datapath optimized to execute one or many MACs in only one cycle.
  - Instruction fetching and decoding overhead is removed
  - Memory access is limited by directly processing the input dataflow

Domain specific processors

- DSPs:
- Designed for high-performance, repetitive,
- numerically intensive tasks
- In one Instruction Cycle, can do:
  - many MAC-operations
  - many memory accesses
  - special support for efficient looping
- The hardware contains:
  - One or more MAC-Units
  - Multi-ported on-chip and off-chip memories
  - Multiple on-chip busses
  - Address generation unit supporting addressing modes tailored for DSP-applications



Application specific processors

- Optimize the complete circuit for a given function
- Example: ASIC: Application Specific Integrated Circuit.
  - Optimization is done by implementing the inherent parallel structure on a chip
  - The data path is optimized for only one application.
  - Instruction fetching and decoding overhead is removed
  - Memory access is limited by directly processing the input data flow
  - Exploitation of parallel computation

Application specific processors

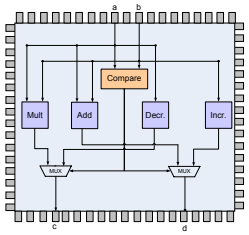
- ASIC Example:
- *Implementation of a VN computer*

```
if (a < b) then
{
    d = a+b;
    c = a*b;
}
else
{
    d = a+1;
    c = b-1;
}
```

- *At least 3 instructions*
- *run-time >= $3*t_{instruction}$*

The VN computer needs to be clocked at least 5 time faster

ASIC implementation:
The complete execution is done in parallel in one clock cycle
run-time = t_{clock} = delay longest path from input to output



Conclusion

- Von Neumann computer:
  - General purpose, used for any kind of function.
    - High degree of flexibility.
  - *However, high restrictions on the program coding and execution scheme*
    - the program have to adapt to the machine
- DSPs are Adapted for a class of applications.
  - Flexibility and efficiency only for a given class of applications.
- ASICs are
  - Tailored for one application.
    - Very efficient in speed and resource.
  - Cannot re-adapt to a new application
    - Not flexible

Reconfigurable Computing

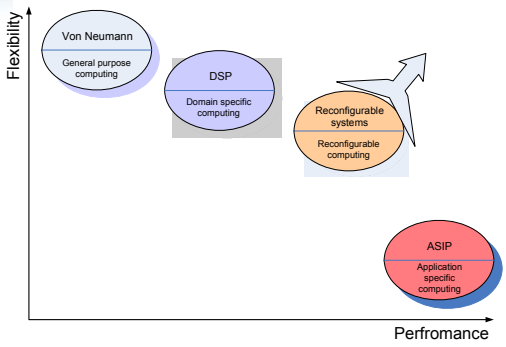
- The Ideal device should combine:
  - the flexibility of the Von Neumann computer
  - the efficiency of ASIPs
- The ideal device should be able to
  - Optimally implement an application at a given time
  - Re-adapt to allow the optimal implementation of a new application.
- We call such a device a reconfigurable device.

Definition: Reconfigurable computing can be defined as the study of computations involving reconfigurable devices. This includes, architecture, algorithms and applications.

Comparison

Technology	Performance	System Design Cost	Cost/unit	Power Consumption	Flexibility
GPP	Low	Low	Low-Medium	High	High
DSP	Medium	Medium	Low-Medium	Medium	Medium
ASIC	High	High	Low	Low	Low
Reconfigurable HW	Medium-High	Medium	Medium-high	Medium-High	High

Flexibility vs Efficiency



Some Fields of Application

- Rapid prototyping
- Post fabrication customization
- Multi-modal computing tasks
- Adaptive computing systems
- Fault tolerance
- High performance parallel computing

Rapid prototyping

- Testing hardware in real conditions before fabrication
  - Software simulation
    - Relatively inexpensive
    - Slow
    - Accuracy ?
  - Hardware emulation
    - Hardware testing under real operation conditions
    - Fast
    - Accurate
    - Allow several iterations


APTIX System Explorer

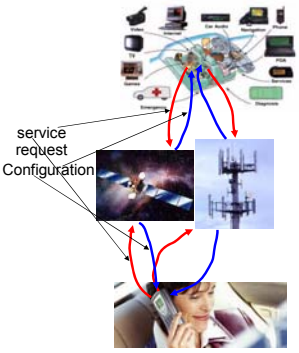

ITALTEL FLEXBENCH

Post fabrication customization

- Time to market advantage
  - Ship the first version of a product
  - Remote upgrading with new product versions
  - Remote repairing

Multi-modal computing tasks

- Reconfigurable vehicles, mobile
  - phones, etc..
  - Built-in Digital Camera
  - Video phone service
  - Games
  - Internet
  - Navigation system
  - Emergency
  - Diagnostics
  - Different standard and protocols
  - Monitoring
  - Entertainment



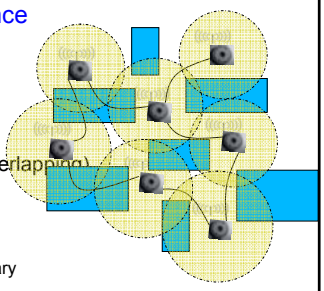
Adaptive computing systems

Computing systems that are able to adapt their behaviour and structure to changing operating and environmental conditions, time-varying optimization objectives, and physical constraints like changing protocols, new standards, or dynamically changing operation conditions of technical systems.

- Dynamic adaptation to environment
- Dynamic adaptation to threats (DARPA)
- Extended mission capabilities

Adaptive Distributed Video Processing

- Application in surveillance
  - Distributed cameras
    - Intelligent
    - Adaptive
    - Performance
  - Each camera covers a given area (Can be overlapping)
  - Communication
    - Data exchange
      - Knowledge
      - Information at boundary
    - Wireless
  - Self-organization
    - Repositioning for better coverage

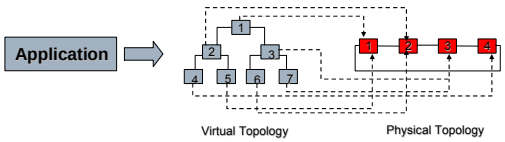


Adaptive Distributed Video Processing

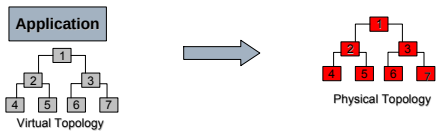
- Operation in normal mode
  - Image understanding
    - Movement detection and tracking
  - Fusion of information
    - Better coverage of a complete area through self-organization
  - Data transmission
    - Characteristics of a suspect in the covering range
- Operation on failure
  - Failure detection mechanism
  - Self healing mechanism
  - Failure recovery
  - Detection and protection against attacks

High performance parallel computing

Traditional parallel implementation flow



Exploiting reconfigurable topology



The main question

- Since a reconfigurable device is a piece of hardware and since a hardware can never change after fabrication
- How is a reconfigurable device made ?

→ More in this in the next chapter

Course Coverage

- Introduction
  - The Von-Neumann computation paradigm
  - Application specific processors
  - Reconfigurable computing
- Reconfigurable Architectures
  - Early work
  - Programmable Logic
    - PAL, PLAs, CPLDs
    - FPGAs, Hybrid FPGAs
  - Coarse grained reconfigurable devices

Course Coverage

- Design and Implementation
  - In-System integration
  - Design Flow
  - Logic synthesis
  - Technology Mapping
  - (Re)configuration
- High-Level Synthesis for RC
  - Temporal partitioning
  - List-scheduling approach
  - Integer Linear Programming
  - Network Flow
  - Spectral methods
  - Iterative improvements

Course Coverage

- Temporal Placement
  - Operating system concepts
  - The temporal placement problem
  - Off-line approaches
  - On-line approaches
- On-line Communication paradigms
  - Communication over a third party
  - Circuit switching
  - Network on Chip (NoC)
  - Dynamic Network on Chip (DyNoC)

Course Coverage

- Applications and existing platforms
  - Pattern Matching
  - Network packet processing
  - Multimedia processing
  - Multi-controller
- SoPC (System on Programmable Chip)
  - Introduction to SoPC
  - Case study: The Xilinx EDK
  - Adaptive Multiprocessing on Chip
- Designing for partial reconfiguration
  - Partial reconfiguration design on the Xilinx platform

Course Coverage

- Project proposal
  - Network packet processing
  - Image processing
  - Reconfigurable gaming
  - Bioinformatics
  - New ideas are welcome
- Implementation Platforms:
  - Xilinx NetFPGA board
  - Altera DE2 board
  - Xilinx XUP board???
- Programming Language:
  - Verilog HDL
  - C/C++