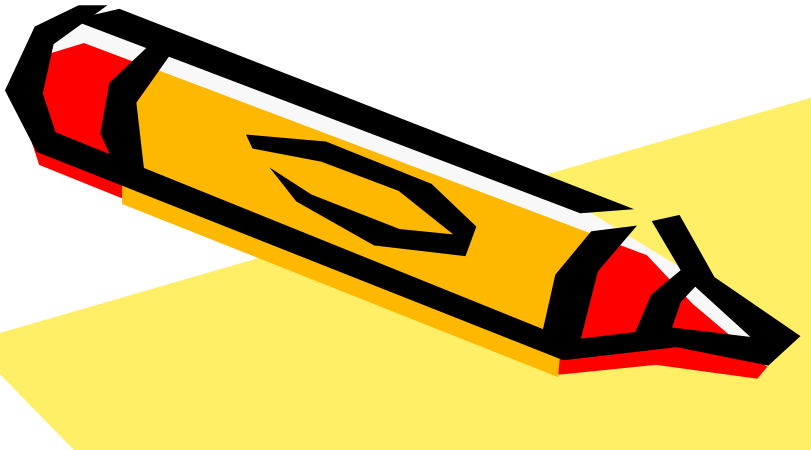


Computer Architecture

Hossein Asadi
Department of Computer Engineering
Sharif University of Technology
asadi@sharif.edu

Today's Topics

- Syllabus & Objectives
- Textbook
- Assignments
- Class Policy
- Reminder from "Logic Design" & "CSL" Course
- Introduction to Computer Architecture

Course Introduction

- Instructor: Hossein Asadi
- Classes
  - Sat & Mon: 9:00AM~10:30AM
  - Attend class on time
- Office Hours
  - Sun. 10:30AM~12:00AM
  - Sun. 4:30PM~6:00PM
  - Wed. afternoons with appointment only
  - Room # 610
- TAs Classes
  - Tuesdays 12:15~1:15PM, Location: 101

Course Introduction (cont.)

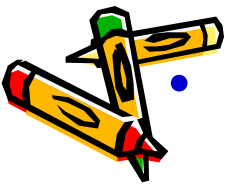
- Course Webpage on CW
  - Check this webpage on regular basis
    - At least on Sun, Tue, Thur
    - Q&A only using CW forums
  - Everything will be posted on CW
    - Announcements, handouts, assignments, grades, quiz and exam notices, simulators, …
  - Handouts
    - Will be posted a day before class
      - Print it & bring it to class
    - But I may update it a day after class
      - Check out submission date of handouts

Copyright Notice

- Parts (text & figs) of lectures adopted from
  - Computer Organization & Design, The Hardware/Software Interface, 4th Edition, by D. Patterson and J. Hennessey, MK publishing, 2012.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, CMU, Spring 2009.
  - "Computer Architecture & Engineering" handouts, by Prof. Kubiatowicz, UC Berkeley, Spring 2004.
  - "Intro to Computer Architecture" handouts, by Prof. Hoe, UWisc, Spring 2019.
  - "Computer Arch I" handouts, by Prof. Garzarán, UIUC, Spring 2009.

Few Notes on Assignments

- Post All your Questions on CW Forums
  - Check forum history before posting any question
- Be Respectful to your Classmates and TAs
- Harsh Cheating Penalty



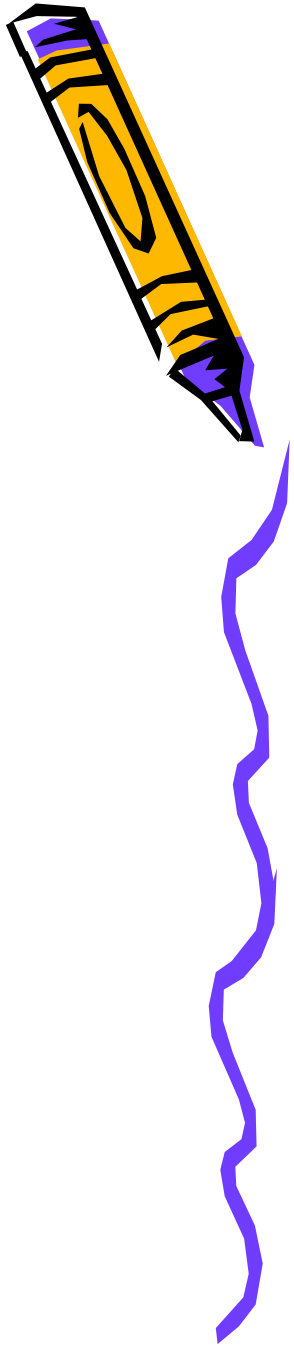
Course Introduction (cont.)

- Course Webpage
  - Sharif CW webpage, http://cw.sharif.edu
  - Make sure to have an account on CW
  - Check this webpage on regular basis
    - At least on Sun, Tue, Thur
  - Everything will be posted online
    - Announcements, assignments, and toolsets
  - Handouts (in pdfs)
    - Print it & bring it to class
    - I may update it a day after class
      - Check out submission date of handouts

Course Introduction (cont.)

- Textbook
  - Computer Organization & Design, The Hardware/Software Interface, 4th Edition, by D. Patterson and J. Hennessey, MK publishing, 2012.

Syllabus



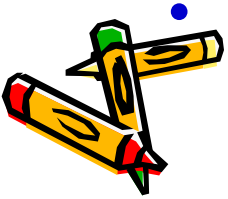
- **Review**
  - Combinational & sequential logic design
  - Design abstractions
  - Computer/CPU history
  - Computer organization
  - Addressing modes
  - Instruction Set Architecture (ISA)
- **Number Representation**
  - Fixed-point
  - IEEE 754 Floating-point standard
    - Single precision and double precision

Syllabus (cont.)

- **Performance Evaluation**
  - Performance
  - Important factors in performance
  - Benchmarks
- **Data-Path and Control-Path Design**
  - Register Transfer Logic (RTL)
  - Data-path components
  - Control unit design and hardwired controller
  - MIPS data-path
  - Interrupt and I/O polling

Syllabus (cont.)

- Micro-Programmed Controller
  - Pros & cons compared to hardwired
- Multi-Cycle Architecture
- Introduction to Pipeline Architecture
- I/O Approaches
  - I/O handshaking
- Introduction to Multi-Core Systems
- Introduction to Parallel Computing



Syllabus (cont.)

- Memory System
  - Types of memory
  - Memory hierarchy
  - Cache memory and cache configurations
- Arithmetic Algorithms
  - Addition, subtraction, multiplication, division
  - Arithmetic architectures
  - Booth and array multiplication

Objective

- Understand Basic Architecture of CPUs
- Be Able to Evaluate and Analyze Performance of Different Processors
  - Using simulation tools
- Understand Arithmetic Algorithms
- Understand Memory Hierarchy
  - And its impact on overall performance
- Understand Basics of Pipelining and Multi-Cores Systems

Objective (cont.)

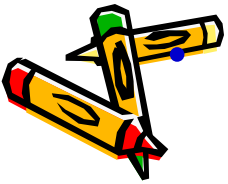
- By the end of semester, you should be able to answer these questions:
  - What is functionality of main components of a processor?
  - Why standard benchmarks used for performance evaluation?
  - What are pros and cons of single-cycle, multi-cycle, and pipelined data-paths?
  - Difference between micro-programmed controller and hardwired controller?

Objective (cont.)

- By the end of semester, you should be able to answer these questions:
  - Tradeoffs of small vs. large L1 caches?
  - How many levels in a cache hierarchy?
  - What are pros and cons of direct-mapped, set-associative, and fully-associative cache configurations?
  - What are pros and cons of different adder implementations (RC, CSA, CLA)?
    - Ripple-carry, carry-select, carry look-ahead adder

Grading

- Midterm Exam: 25%
  - Farvardin 24th
- Final Exam: 30% (date posted in EDU)
- Quiz (1&2): 15%
  - First quiz: Esfand 20th
  - Second quiz: Ordibehesht 21st
  - Up to three additional unscheduled quizzes
- Assignments & Project: 30%
  - Bonus points for outstanding projects
- Exams: Topics of this Class and TA Classes



Class Policy

- Ask Questions Anytime
  - Don't hesitate to ask even stupid questions!!!
- Cell Phones Off or on Silent
- Absence
  - Only three sessions allowed
- Food No, Drink yes!
- Feel Free to Pass Me Your Feedbacks
  - Anything related to this course

Assignments

- 10~12 Assignments
  - 5~6 analytical assignments
  - 5~6 design & simulation assignments
    - Altera Quartus toolset ©
    - SimpleScalar toolset ©
  - Spend enough time on assignments as they will be covered in midterm and final exams

Assignments (cont.)

- Assignment Policy
  - Two late assignments will be accepted!
    - Only two days late!
    - Third late assignment (two-day late)
      - HW will be graded out of 50%
    - Forth and next late assignments will not be accepted!
  - Discussions encouraged!
  - But do your own handwriting!
  - Zero score for copied assignments!
    - Second time zero score for 30% share!

What You Learned So Far

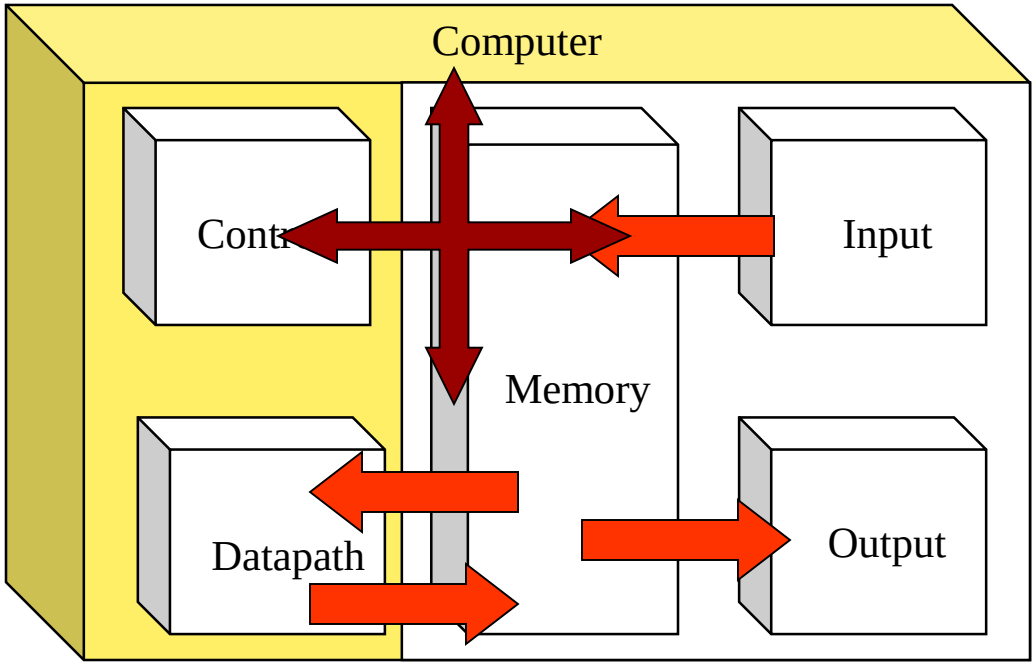
- Logic Design
  - Simple logical & arithmetic logic design
    - Addition and subtraction units
    - Multiplexer and tri-state buffer
    - Latch and flip-flop
    - Sequential logic, registers, shifters, counters
- Computer Structure & Language
  - Computer organization
  - Instruction Set Architecture (ISA)
  - Assembly programming
- Now "Computer Architecture"
  - What is "Computer Architecture"?

Reminder: Computer Systems

- A computer system consists of hardware and software that are combined to provide a tool to solve problems (with best performance)
  - Hardware may include:
    - CPU, memory, disks, printers, screen, keyboard, mouse, …
  - Software may include:
    - System software
      - A general environment to create specific applications
    - Application software
      - A tool to solve a specific problem

Reminder: Computer Organization

- Computer Components
  - Input, output, memory, control unit, & datapath



Reminder: ISA

- Instruction Set Architecture (ISA)
  - A set of instructions used by a machine to run programs
  - Interface between hardware & software
  - Provides an abstraction of hardware implementation
    - Hardware implementation decides what and how instructions are implemented
  - ISA specifies
    - Instructions, Registers, Memory access, Input/output

Reminder: ISA (cont.)

- Key ISA Decisions
  - Instruction length?
  - How many registers?
  - Where operands reside?
    - Which instructions can access memory?
  - Instruction format?
  - Operand format?
    - How many? How big?

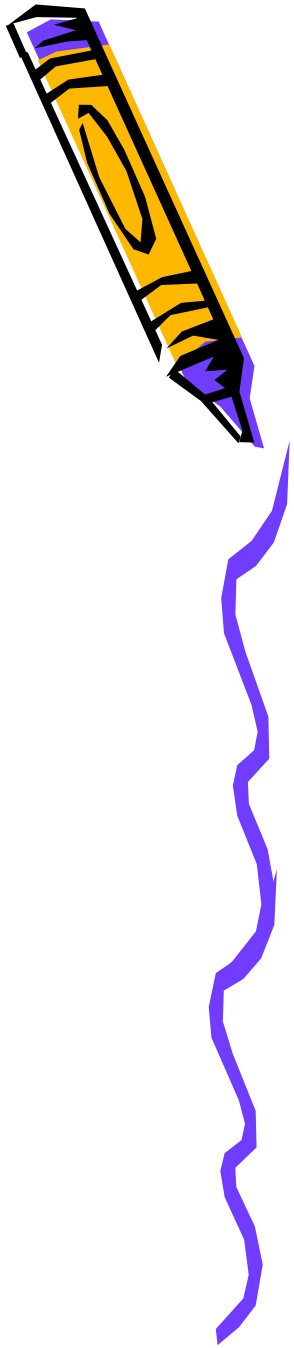
Reminder: ISA (cont.)

- ISA Classes

Code sequence for `C = A + B`

Stack	Accumulator	Register-Memory	Load-Store
`Push A`	`Load A`	`Add C, A, B`	`Load R1,A`
`Push B`	`Add B`		`Load R2,B`
`Add`	`Store C`		`Add R3,R1,R2`
`Pop C`			`Store C,R3`

Reminder: Addressing Modes



- Addressing Modes
  - Immediate addressing
  - Register addressing
  - Base or displacement addressing
  - PC-relative addressing
  - Pseudo-direct addressing
  - Register indirect
  - Direct
  - Memory indirect
  - Scaled
  - Auto-increment / Auto-decrement
  - Indexed

Computer Systems Abstractions

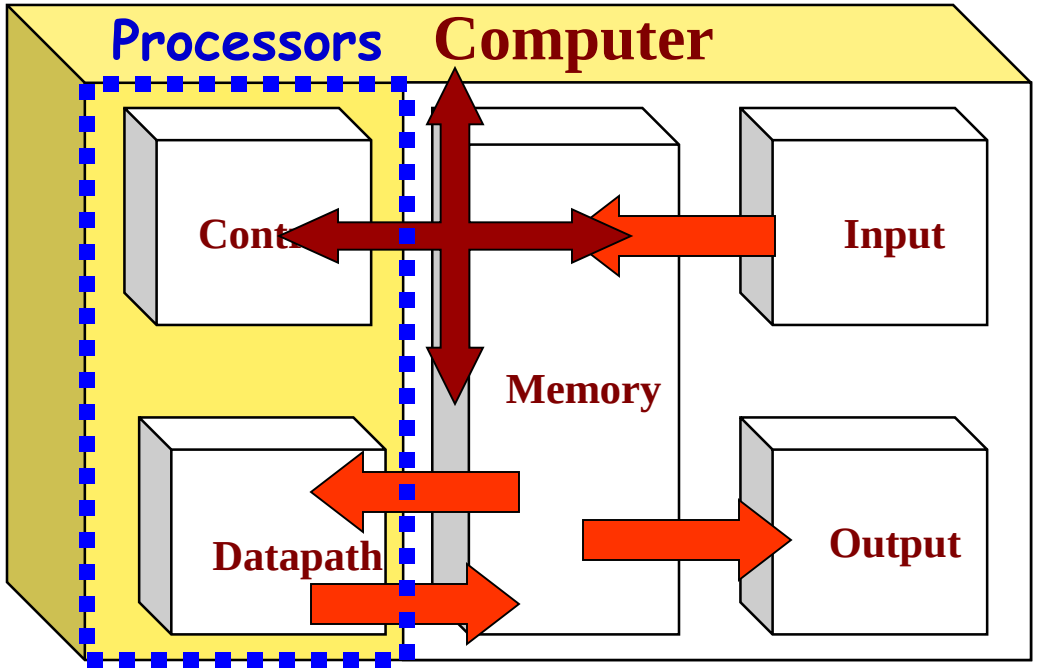
Applications	User and problems
Compilers	Prog. Languages
Operating System (OS)	Resources / virtualization
Architecture (ISA)	HW/SW interface
Micro-architecture	Datapath
Digital Design	Registers, ALU
Circuit	Digital logic
Device	Transistors, signals
	Atoms, electrons

Micro-Architecture (uArch)

- Definition from Wiki
  - A way a given ISA is implemented on a processor
- ISA
  - Can be implemented with different uArch
  - Why different implementation?
    - Different goals (performance, power, cost, …)
- Computer Architecture?
  - ISA + uArch

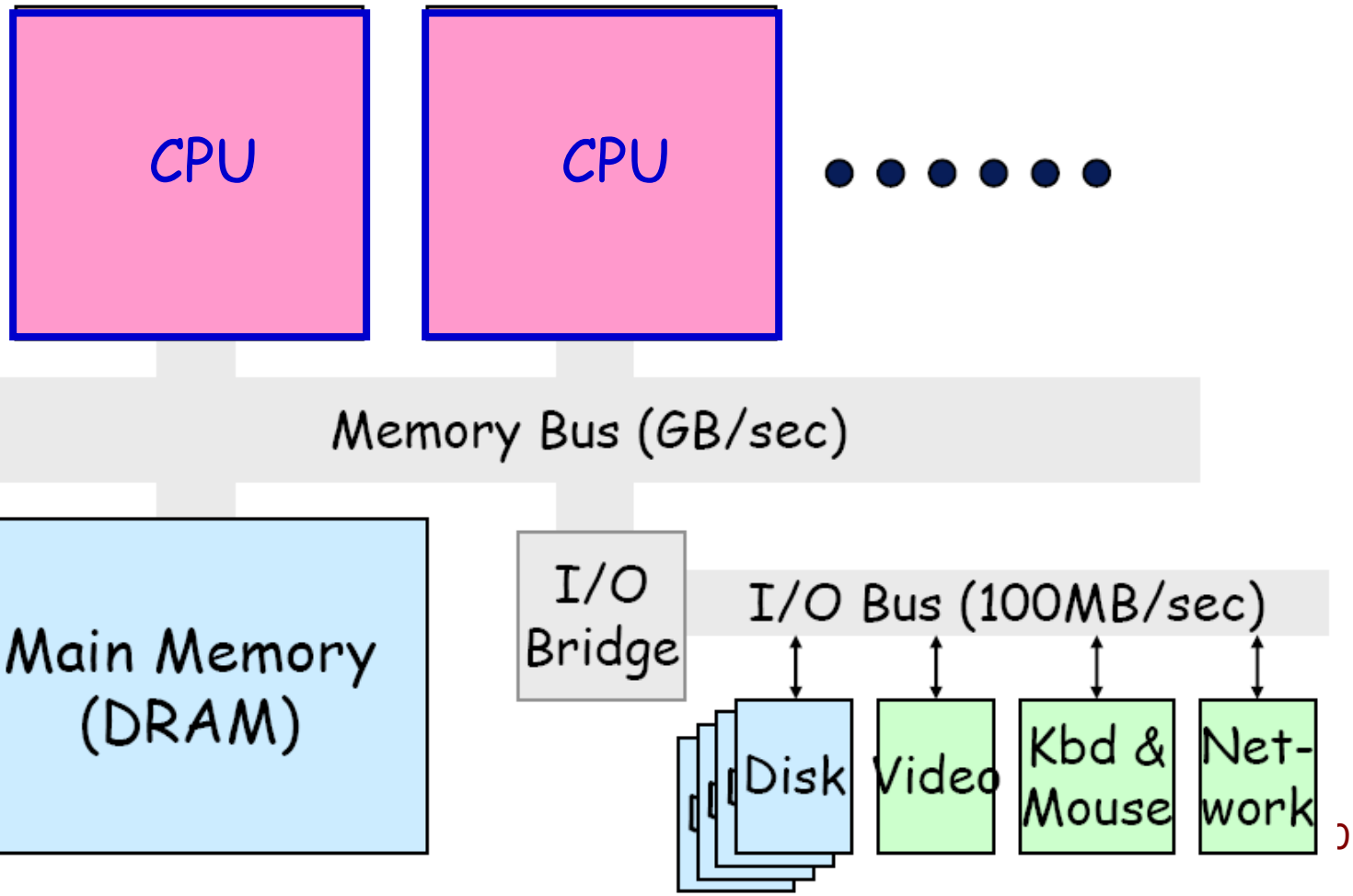
Computer Organization

- Computer Components
  - Input, output, memory, control unit, & datapath



Computer Organization

- Computer Components



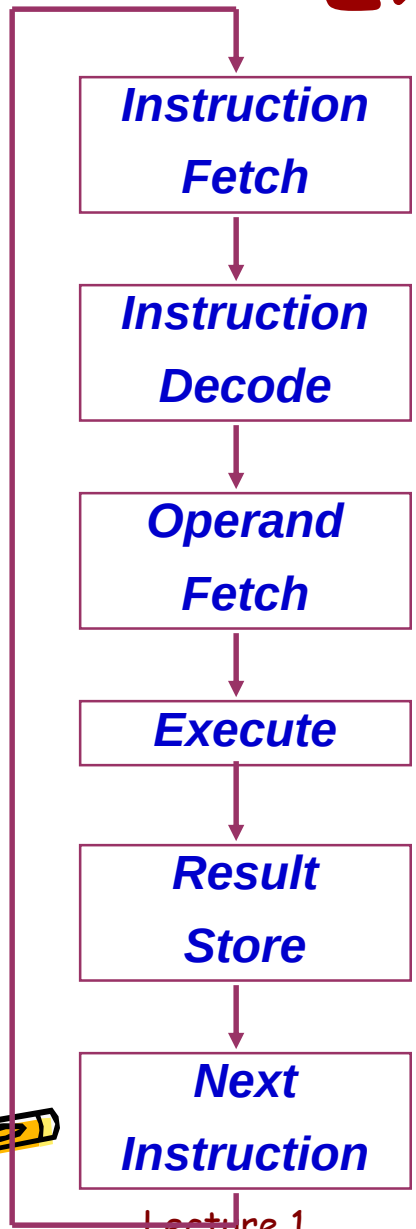
Typical ISA

- Data Transfer Instructions
  - CPU ⇔ Memory
  - CPU ⇔ I/O
- Arithmetic & Logical Instructions
- Control Instruction
  - Conditional branch
  - Unconditional branch

Reminder: Von-Neumann Model

- Stored Program
  - Instructions stored in a linear memory array
- Sequential Instruction Processing
  1. Program counter identifies current instruction
  2. Instructions fetched one by one from memory
  3. Once fetched, instruction is executed
  4. Results stored in memory
  5. Program counter incremented
  6. Return to step 1

Execution Cycle



Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

Compute result value or status

Deposit results in storage for later use

Determine successor instruction

Micro-Architecture

- BIG Picture
  - Basic blocks
    - Components need to execute Von-Neumann algorithm

Micro-Architecture

- Basic Blocks of a Micro-Architecture
  - A high-speed unit to keep code & data
    - CPU runs very fast but memory is slow
    - Cache memory (instruction & data cache)
  - A unit to fetch instructions from cache
    - Instruction fetch unit (IFU)
    - Instructions transferred from I-cache to IFU
  - A unit to decode instructions after fetch process
    - Instruction decoder unit

Micro-Architecture (cont.)

- Basic Blocks of a Micro-Architecture
  - A unit to execute instructions
    - Execution unit
  - A unit to do arithmetic/logical operations
    - ALU
  - A unit to execute branch instruction
    - Branch unit
  - A unit to execute load/store instructions
    - Load/store unit
    - LSU ⇔ D-cache

Micro-Architecture (cont.)

- Basic Blocks of a Micro-Architecture
  - A unit to save temporary results within processor
    - Register file
  - A unit to locate next instruction
    - Program counter
  - A unit to schedule all data movements
    - Control unit

Backup