Computer Architecture

Prof. Dr. Nizamettin AYDIN

naydin@yildiz.edu.tr
nizamettinaydin@gmail.com

http://www.yildiz.edu.tr/~naydin

MIPS ISA - I

Outline

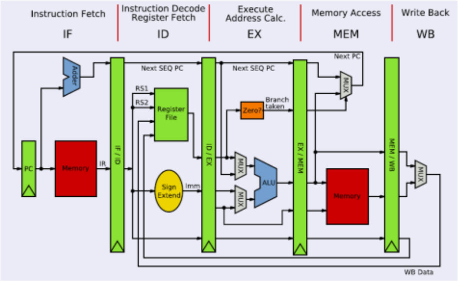
- **Overview of the MIPS architecture**
  - ISA vs Microarchitecture?
  - CISC vs RISC
  - Basics of MIPS
  - Components of the MIPS architecture
  - Datapath and control unit
  - Memory
  - Memory addressing issue
  - Other components of the datapath
  - Control unit

Computer Architecture

- can be viewed as
  - the machine language the CPU implements
    - Instruction set architecture (ISA)
      - Built in data types (integers, floating point numbers)
      - Fixed set of instructions
      - Fixed set of on-processor variables (registers)
      - Interface for reading/writing memory
      - Mechanisms to do input/output
  - how the ISA is implemented
    - Microarchitecture

Computer Architecture

- Microarchitecture



MIPS Architecture

- MIPS
  - An acronym for *Microprocessor without Interlocking Pipeline Stages*
  - Not to be confused with a unit of computing speed equivalent to a *Million Instructions Per Second*
- MIPS processor
  - born in the early 1980s from the work done by John Hennessy and his students at Stanford University
    - exploring the architectural concept of RISC
  - Originally used in Unix workstations,
  - now mainly used in small devices
    - Play Station, routers, printers, robots, cameras

MIPS Architecture

Designer	MIPS Technologies, Imagination Technologies
Bits	64-bit (32 → 64)
Introduced	1985;
Version	MIPS32/64 Release 6 (2014)
Design	RISC
Type	Register-Register
Encoding	Fixed
Branching	Compare and branch
Endianness	BigEndian
Page size	4 KB
Extensions	MDMX, MIPS-3D
Open	No
Registers	
– General purpose	32
– Floating point	32

- Source: wikipedia

CISC vs RISC

- Advantages of CISC Architecture:
  - Microprogramming is easy to implement and much less expensive than hard wiring a control unit.
  - It is easy to add new commands into the chip without changing the structure of the instruction set as the architecture uses general-purpose hardware to carry out commands.
  - This architecture makes the efficient use of main memory since the complexity (or more capability) of instruction allows to use less number of instructions to achieve a given task.
  - The compiler need not be very complicated, as the microprogram instruction sets can be written to match the constructs of high level languages.

CISC vs RISC

- Disadvantages of CISC Architecture:
  - A new or succeeding versions of CISC processors consists early generation processors in their subsets.
    - Therefore, chip hardware and instruction set became complex with each generation of the processor.
  - The overall performance of the machine is reduced due to the different amount of clock time required by different instructions.
  - This architecture necessitates on-chip hardware to be continuously reprogrammed.
  - The complexity of hardware and on-chip software included in CISC design to perform many functions.

CISC vs RISC

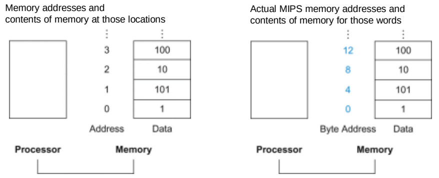
- Advantages of RISC Architecture:
  - The performance of RISC processors is often two to four times than that of CISC processors because of simplified instruction set.
  - This architecture uses less chip space due to reduced instruction set.
    - This makes to place extra functions like floating point arithmetic units or memory management units on the same chip.
  - The per-chip cost is reduced by this architecture that uses smaller chips consisting of more components on a single silicon wafer.
  - RISC processors can be designed more quickly than CISC processors due to its simple architecture.
  - The execution of instructions in RISC processors is high due to the use of many registers for holding and passing the instructions as compared to CISC processors.

CISC vs RISC

- Disadvantages of RISC Architecture:
  - The performance of a RISC processor depends on the code that is being executed.
    - The processor spends much time waiting for first instruction result before it proceeds with next subsequent instruction, when a compiler makes a poor job of scheduling instruction execution.
  - RISC processors require very fast memory systems to feed various instructions.
    - Typically, a large memory cache is provided on the chip in most RISC based systems.

MIPS Basics

- Instructions
  - 4 bytes (32 bits)
  - 4-byte aligned (i.e., they start at addresses that are a multiple of 4 --0x0000, 0x0004, etc.)


Memory addresses and contents of memory at those locations
3
2
1
0
100
10
101
1
Address
Data
Processor
Memory
Actual MIPS memory addresses and contents of memory for those words
12
8
4
0
100
10
101
1
Byte Address
Data
Processor
Memory

  - Instructions operate on memory and registers

Bytes and WordsAddressData

- In modern ISAs memory is byte addressable

Byte addresses

Address	Data
0x0000	0xAA
0x0001	0x15
0x0002	0x13
0x0003	0xFF
0x0004	0x76
...	.
0xFFFE	.
0xFFFF	.

Half Word Addresses

Address	Data
0x0000	0xAA15
0x0002	0x13FF
0x0004	.
0x0006	.
...	.
...	.
...	.
0xFFFC	.

Word addresses

Address	Data
0x0000	0xAA1513FF
0x0004	.
0x0008	.
0x000C	.
...	.
...	.
...	.
0xFFFC	.

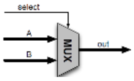
- In MIPS, half words and words are aligned.

MIPS Basics

- Memory Data types (also aligned)
  - Bytes (8 bits)
  - Half words (16 bits)
  - Words (32 bits)
  - Memory is denoted as M
    - e.g., M[0x10] is the byte at address 0x10
- Registers
  - 32 4-byte registers in the register file
  - Denoted as R
    - e.g., R[2] is register 2

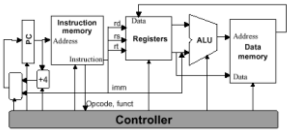
Components of the MIPS architecture

- Major hardware components of the instruction cycle
  - Datapath
    - path of instructions and data through the processor
    - components connected by buses
      - [Bus- parallel path for transmitting values]
      - [in MIPS, usually 32 bits wide]
  - Control unit
    - controls the components of the datapath
    - determines how data moves through the datapath
    - receives condition signals from the components
    - sends control signals to the components
    - switches between buses with multiplexers



Components of the MIPS architecture

- Major components of the datapath:
  - program counter (PC)
  - instruction register (IR)
  - register file
  - arithmetic and logic unit (ALU)
  - memory
- Control unit



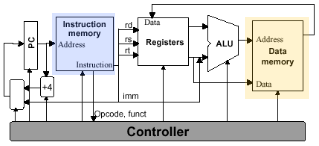
http://www.cise.ufl.edu/~mssz/CompOrg/Figure4.3-MIPSarch2.gif

Components of the MIPS architecture

- Memory
    - In MIPS, programs are separated from data in memory
  - Text segment
    - instruction memory
    - part of memory that stores the program (machine code)
    - read only
  - Data segment
    - data memory
    - part of memory that stores data manipulated by program
    - read/write

Components of the MIPS architecture

- Memory



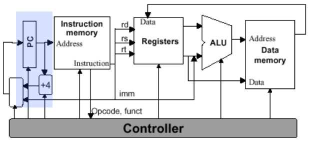
- Distinction may or may not be reflected in the hardware:
  - von Neumann architecture (single, shared memory)
  - Harvard architecture (physically separate memories)

Components of the MIPS architecture

- Memory addressing in MIPS
  - Memory address computed as base+offset for reading/writing the data segment
    - base is obtained from a register
    - offset is given directly as an integer
  - Load word (read word from memory into register):
    lw $t1, 8($t2) ➔ $t1 := Memory[$t2+8]
  - Store word (write word from register into memory):
    sw $t1, 4($t2) ➔ Memory[$t2+4] := $t1

Components of the MIPS architecture

- Program counter (PC)
  - Register that stores the address of the next instruction to fetch



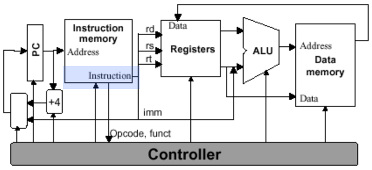
- also called the instruction pointer (IP)

- Program
  - a sequence of machine instructions in the text segment

```
0x8d0b0000
0x8d0c0004
0x016c5020
0xad0a0008
0x21080004
0x2129ffff
0x1d20fff9
```

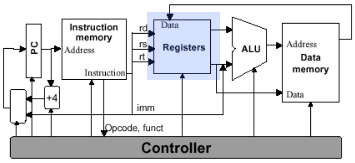
Components of the MIPS architecture

- Instruction register (IR)
  - Register that holds the instruction currently being decoded



Components of the MIPS architecture

- Register file
  - MIPS register file contains 32 registers



- Register
  - component that stores a 32-bit value

The MIPS Register File

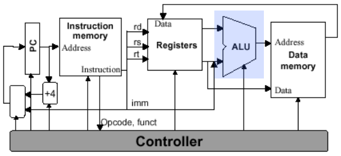
- All registers are the same
  - where a register is needed any register will work

Name	number	use	Callee saved
$zero	0	zero	n/a
$at	1	Assemble Temp	no
$v0 - $v1	2 - 3	return value	no
$a0 - $a3	4 - 7	arguments	no
$t0 - $t7	8 - 15	temporaries	no
$s0 - $s7	16 - 23	saved temporaries	yes
$t8 - $t9	24 - 25	temporaries	no
$k0 - $k1	26 - 27	Res. for OS	yes
$gp	28	global ptr	yes
$sp	29	stack ptr	yes
$fp	30	frame ptr	yes
$ra	31	return address	yes

- By convention, we use them for particular tasks
  - Argument passing
  - Temporaries, etc.
- $zero is the zero register
  - It is always zero.
  - Writes to it have no effect.

Components of the MIPS architecture

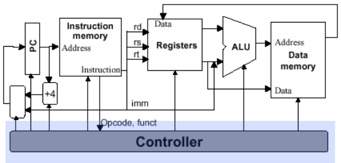
- Arithmetic and logic unit (ALU)
  - Implements binary arithmetic and logic operations



- Inputs:
  - operands (2×32-bit)
  - operation (control signals)
- Outputs:
  - result (1×64-bit)
  - status (condition signals)

Components of the MIPS architecture

- Control unit
  - Controls components of datapath to implement instruction cycle
  - Implemented as a finite state machine

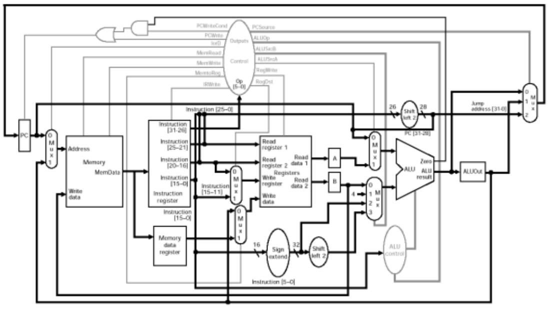


- Inputs:
  - condition signals
- Outputs:
  - control signals

Components of the MIPS architecture

- Control unit
  - Condition signals
    - from IR – decode operation, arguments, result location
    - from ALU – overflow, divide-by-zero, . . .
  - Control signals
    - to multiplexors – buses to select
    - to each register – load new value
    - to ALU – operation to perform
    - to all – clock signal
      - clock signal
        - ensures that outputs are not used until ready
        - synchronizes the components in the data path
      - the faster the clock, the faster the program executes
        - clock rate is limited by the slowest component!

MIPS data path with control signals



http://fourier.eng.hmc.edu/e85/lectures/figures/MIPS_datapath_control.gif