

CPS104
Review of Important Course Topics

Robert Wagner

CPS104 Rev.1 ©RW Fall 2000

Lecture 2 : C and C++: Computer Memory

- Memory is a large linear array of 8 bit bytes.
 - * Each byte has a unique address (location)
- Bytes are grouped into longer sequences, some of which can be addressed as one unit. Their addresses must be multiples of their byte lengths.
 - * Byte (1 byte) -- characters (char)
 - * Word (4 bytes) -- integers (int), floats.
 - * Double (8 bytes)

Address	Data
0	01010110
1	10001100
2	
3	
4	
...	
2 ⁿ -1	

CPS104 Rev.2 ©RW Fall 2000

Lecture 3 : Data representations

- Introduction: Decimal, Binary, Octal and Hexadecimal numbers.
- Storage types: Byte, Word, Double-word.
- ASCII Characters.
- 2's complement numbers.
- Floating-point numbers.
- Computer Instructions.
- Memory Addresses.

Sections 3,7, 4.1-4.3, 4.8 In the textbook

CPS104 Rev.3 ©RW Fall 2000

Lecture 6: MIPS ISA and Assembler :

- The MIPS Assembly Language.
- MIPS Assembly Language Programming Conventions.
- The program Stack
- Useful C techniques: “case” selection, “hash lookup”

- ★ Reading Assignment: Chapter 3, Appendix A
- ★ SPIM manual.

CPS104 Rev.4

©RW Fall 2000

Problem P6:
MYMIPS: Instruction Set Architecture, Simulator:
 Lecture 12: A MYMIPS CPU Data Path:
MYMIPS ISA Subset

OP	Name	Action
31..28		
0001	Load	R[D]=MEM[R[A]+E()]; PC = PC+4
0011	Store	MEM[R[A]+E()]=R[D]; PC = PC+4
0101	Add	R[D]=R[A] + E(); PC = PC+4
0111	And	R[D]=R[A] & E(); PC = PC+4
1110	BrCond	PC = (T(D,R[A]) ? E() : PC+4

D=(29:24), A=(23:20), I=(19), IMM=(18:0), B=(3:0)

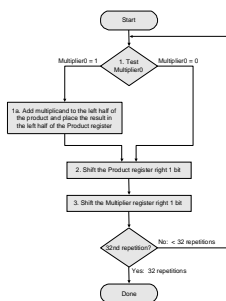
E() = I ? SignExt(IMM) : R[B]; EU() = I ? IMM : R[B]

D:	0	1	2	3	4	5	6	7
T(D,X):	1	X<0	X==0	X<=0	X>0	X!=0	X>=0	0

CPS104 Rev.5

©RW Fall 2000

Lecture 9: Integer Arithmetic: **Multiplication Algorithm #2**



CPS104 Rev.6

Figure Copyright Morgan Kaufmann
 ©RW Fall 2000

Lecture 10: Boolean Algebra & gates:

- Truth tables, Boolean functions, Gates and Circuits
- Karnaugh maps for simplifying Boolean equations
- Examples: 2-1 MUX, Full Adder

Read Appendix B

CPS104 Rev.7 ©RW Fall 2000

Lecture 11: Gates, Buses, Latches:

- The MIPS ALU
- Shifter
- The Tristate driver
- Bus Interconnections
- Register Cell
- The Register File

Read Appendix B

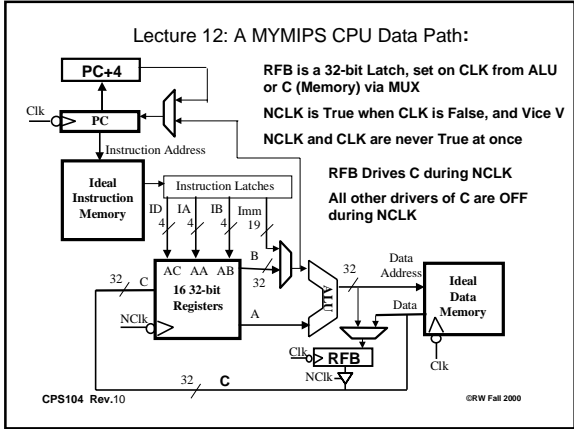
CPS104 Rev.8 ©RW Fall 2000

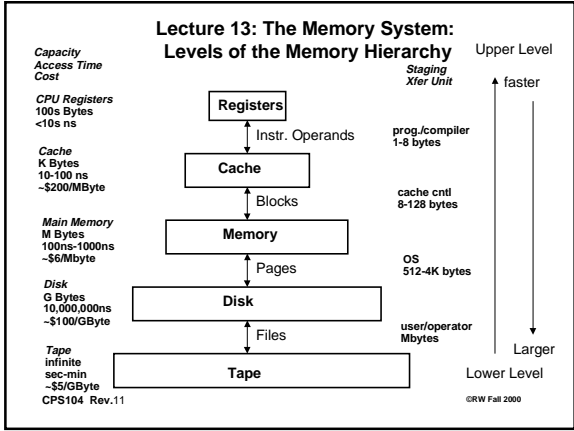
Lecture 12: A MYMIPS CPU Data Path:

- Designing a CPU
 - * From description of ISA (MYMIPS subset) to Hardware "Program"
 - * Note patterns common to many instructions
 - Develop signals to indicate when these occur
 - * Specify major CPU components
 - ALU, Register file, Memory must be tailored to MYMIPS
 - * Overall CPU operation
 - * Specify circuits using shorthand notation (RTL)
 - * Translate RTL to circuits
 - BX signal computation
 - MYMIPS ALU, incl bi-directional shifter
 - Tailored Register File
 - * Examine overall design for flaws

Read Appendix B

CPS104 Rev.9 ©RW Fall 2000





- ### Lecture 14: The Cache
- The Memory Hierarchy
 - Direct Mapped Cache.
 - Two-Way Set Associative Cache
 - Fully Associative cache
 - Replacement Policies
 - Write Strategies
- CPS104 Rev.12 ©RW Fall 2000

Lecture 16: Virtual Memory

- Virtual Memory.
 - * Paged virtual memory.
 - * Virtual to Physical translation: The page table.
 - * Fragmentation.
 - * Page replacement policies.
 - * Reducing the Virtual to Physical address translation time.
 - The TLB
 - Parallel access to the TLB and Cache
 - * Memory Protection
 - * Putting it all together: The SPARC-20 memory system

CPS104 Rev.13

©RW Fall 2000

Lecture 17: Interrupts, Exceptions and Traps

- Interrupts, Exceptions and Traps are asynchronous changes in the control flow. Interrupts and Exceptions can be viewed as asynchronous (unscheduled) procedure calls.
- Interrupts and exceptions are designed to provide:
 - ◆ protection mechanisms: Error handling, TLB management.
 - ◆ efficiency: Overlap I/O and execution, . . .
 - ◆ Illusion of parallelism: Timesharing, multithreading
 - ◆ Ability to handle Asynchronous external events: Network connections, keyboard input, DMA I/O, . . .

CPS104 Rev.14

©RW Fall 2000

Lecture 17: Interrupts, Exceptions and Traps

- **Exception:** a change in execution caused by a condition that occurs within the processor.
 - ◆ segmentation fault (access outside program boundaries, illegal access, . . .)
 - ◆ bus error
 - ◆ divide by 0
 - ◆ overflow
 - ◆ page fault (virtual memory...)
- **Interrupt:** a change in execution caused by an external event
 - ◆ devices: disk, network, keyboard, etc.
 - ◆ clock for timesharing (multitasking)
 - ◆ These are useful events, must do something when they occur.
- **Trap:** a user-requested exception
 - ◆ Operating system call (syscall)
 - ◆ Breakpoints (debugging mode)

CPS104 Rev.15

©RW Fall 2000

Lecture 18: Input-Output

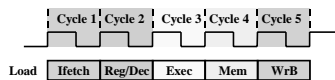
- The I/O system
- Magnetic Disk
- Magnetic Tape
- Buses
- Direct Memory Access

CPS104 Rev.16

©RW Fall 2000

Lecture 19: Pipelining

The Five Stages of Load

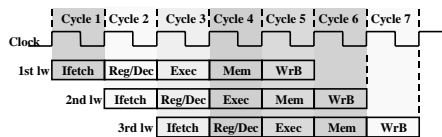


- Ifetch: Instruction Fetch
 - * Fetch the instruction from the Instruction Memory
- Reg/Dec: Register Fetch and Instruction Decode
- Exec: Calculate the memory address
- Mem: Read the data from the Data Memory
- WrB: Write the data back to the register file

CPS104 Rev.17

©RW Fall 2000

Lecture 19: Pipelining Pipelining the Load Instruction



- The five independent functional units in the pipeline datapath are:
 - * Instruction Memory for the Ifetch stage
 - * Register File's Read ports (bus A and bus B) for the Reg/Dec stage
 - * ALU for the Exec stage
 - * Data Memory for the Mem stage
 - * Register File's Write port (bus W) for the WrB stage
- One instruction enters the pipeline every cycle
 - * One instruction comes out of the pipeline (completed) every cycle
 - * The "Effective" Cycles per Instruction (CPI) is 1; $\sim 1/5$ cycle time

CPS104 Rev.18

©RW Fall 2000
