

CSE 490/590 Computer Architecture

ISAs and MIPS

Steve Ko
Computer Sciences and Engineering
University at Buffalo

CSE 490/590, Spring 2011

Last Time...

- Computer Architecture >> ISAs and RTL
- Comp. Arch. shaped by technology and applications
- Computer Architecture brings a quantitative approach to the table
 - 5 quantitative principles of design
- The current performance trend shows that
 - Latency lags behind bandwidth

CSE 490/590, Spring 2011

2

Instruction Set Architecture (ISA)

- The contract between software and hardware
- Typically described by giving all the programmer-visible state (registers + memory) plus the semantics of the instructions that operate on that state
- IBM 360 was first line of machines to separate ISA from implementation (aka. *microarchitecture*)
- Many implementations possible for a given ISA
 - E.g., today you can buy AMD or Intel processors that run the x86-64 ISA.
 - E.g.2: many cellphones use the ARM ISA with implementations from many different companies including TI, Qualcomm, Samsung, Marvell, etc.
 - E.g.3., the Soviets build code-compatible clones of the IBM360, as did Amdahl after he left IBM.

CSE 490/590, Spring 2011

3

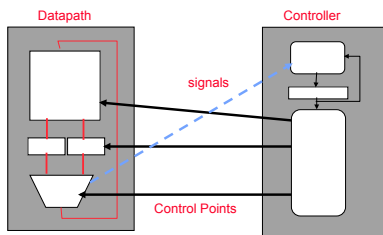
ISA to Microarchitecture Mapping

- ISA often designed with particular microarchitectural style in mind, e.g.,
 - CISC => microcoded
 - RISC => hardwired, pipelined
 - VLIW => fixed-latency in-order parallel pipelines
 - JVM => software interpretation
- But can be implemented with any microarchitectural style
 - Intel Nehalem: hardwired pipelined CISC (x86) machine (with some microcode support)
 - Intel could implement a dynamically scheduled out-of-order VLIW Itanium (IA-64) processor
 - ARM Jazelle: A hardware JVM processor

CSE 490/590, Spring 2011

4

Datapath vs Control

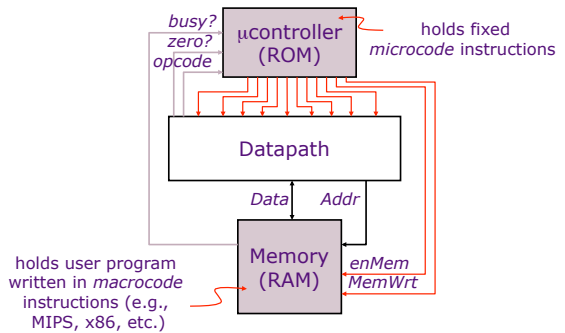


- Datapath: Storage, FU, interconnect sufficient to perform the desired functions
 - Inputs are Control Points
 - Outputs are signals
- Controller: State machine to orchestrate operation on the data path
 - Based on desired function and signals

CSE 490/590, Spring 2011

5

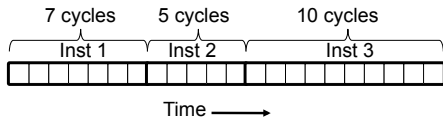
Microcoded Microarchitecture (CISC)



CSE 490/590, Spring 2011

6

CISC

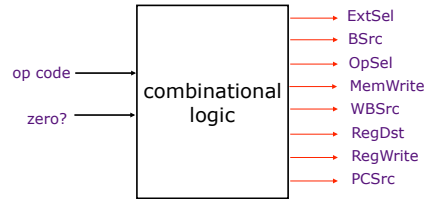


- Variable cycles per instruction
- Variable address modes
 - reg-reg, reg-mem, mem-mem, etc.
- Convenient for programmers
 - Support many instructions
- Difficult to predict completion time
 - No good for pipelining

CSE 490/590, Spring 2011

7

Hardwired Control is pure Combinational Logic (RISC)



CSE 490/590, Spring 2011

8

A "Typical" RISC ISA

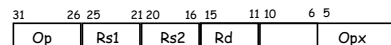
- 32-bit fixed format instruction (3 formats)
- 32 32-bit GPR (R0 contains zero, DP take pair)
- 3-address, reg-reg arithmetic instruction
- Single address mode for load/store:
 - base + displacement
 - no indirection
- Simple branch conditions
- Delayed branch
- Designed for use by compilers & pipelining
 - see: SPARC, MIPS, HP PA-Risc, DEC Alpha, IBM PowerPC, CDC 6600, CDC 7600, Cray-1, Cray-2, Cray-3

CSE 490/590, Spring 2011

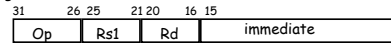
9

Example: MIPS

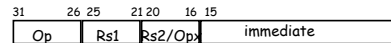
Register-Register



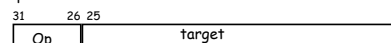
Register-Immediate



Branch



Jump / Call



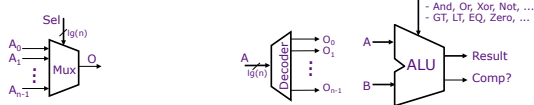
CSE 490/590, Spring 2011

10

Hardware Elements

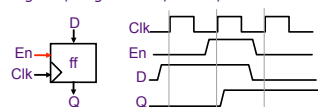
- Combinational circuits

- Mux, Decoder, ALU, ...



- Synchronous state elements

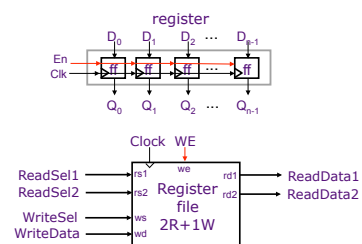
- Flipflop, Register, Register file, SRAM, DRAM



Edge-triggered: Data is sampled at the rising edge

CSE 490/590, Spring 2011

Register Files

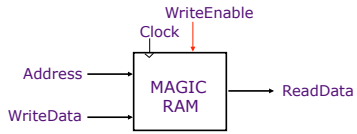


- Reads are combinational

CSE 490/590, Spring 2011

12

A Simple Memory Model



- Reads and writes are always completed in one cycle
- a Read can be done any time (i.e. combinational)
 - a Write is performed at the rising clock edge if it is enabled
 - ⇒ *the write address and data must be stable at the clock edge*

CSE 490/590, Spring 2011

13

CSE 490/590 Administrivia

- Please check the web page: <http://www.cse.buffalo.edu/~stevko/courses/cse490/spring11>
- Don't forget
 - Recitations start from this week.
 - Please purchase a BASYS2 board (100K) as soon as possible.
 - Projects should be done individually.
- Please read the syllabus webpage.
- I have no idea how fast/slow I'm going.
 - Please stop me if too fast!

CSE 490/590, Spring 2011

14

Implementing MIPS: Single-cycle per instruction datapath & control logic

CSE 490/590, Spring 2011

15

The MIPS ISA

Processor State

- 32 32-bit GPRs, R0 always contains a 0
- 32 single precision FPRs, may also be viewed as 16 double precision FPRs
- FP status register, used for FP compares & exceptions
- PC, the program counter
- some other special registers

Data types

- 8-bit byte, 16-bit half word
- 32-bit word for integers
- 32-bit word for single precision floating point
- 64-bit word for double precision floating point

Load/Store style instruction set

- data addressing modes- immediate & indexed
- branch addressing modes- PC relative & register indirect
- Byte addressable memory- big endian mode

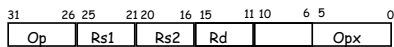
All instructions are 32 bits

CSE 490/590, Spring 2011

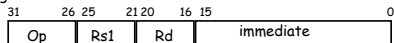
16

Example: MIPS

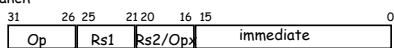
Register-Register



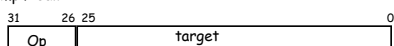
Register-Immediate



Branch



Jump / Call



CSE 490/590, Spring 2011

17

Instruction Execution

Execution of an instruction involves

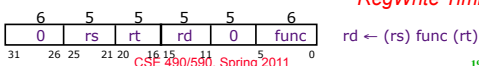
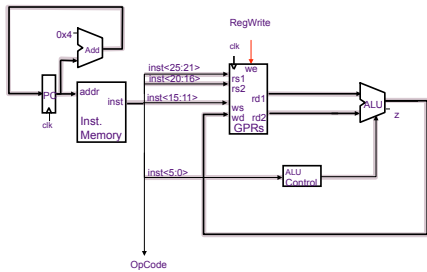
1. instruction fetch
2. decode and register fetch
3. ALU operation
4. memory operation (optional)
5. write back

and the computation of the address of the *next instruction*

CSE 490/590, Spring 2011

18

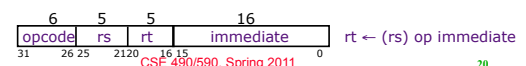
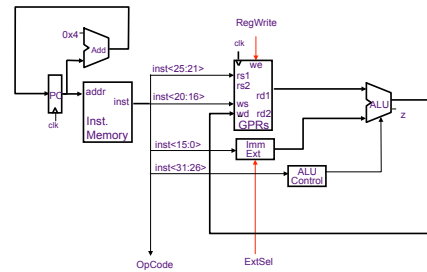
Datapath: Reg-Reg ALU Instructions



CSE 490/590, Spring 2011

19

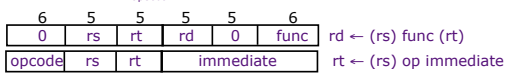
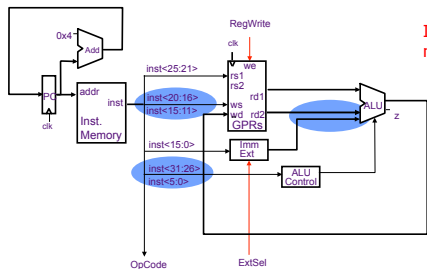
Datapath: Reg-Imm ALU Instructions



CSE 490/590, Spring 2011

20

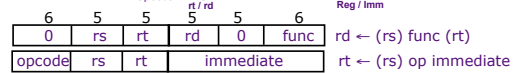
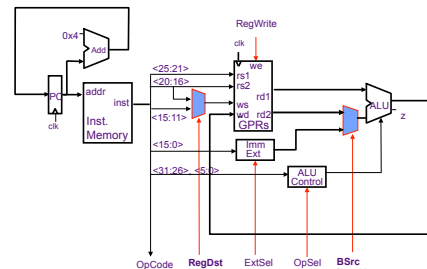
Conflicts in Merging Datapath



CSE 490/590, Spring 2011

21

Datapath for ALU Instructions



CSE 490/590, Spring 2011

22

Datapath for Memory Instructions

Should program and data memory be separate?

Harvard style: separate (Aiken and Mark 1 influence)

- read-only program memory
- read/write data memory

- Note:
Somehow there must be a way to load the program memory

Princeton style: the same (von Neumann's influence)

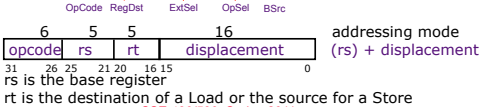
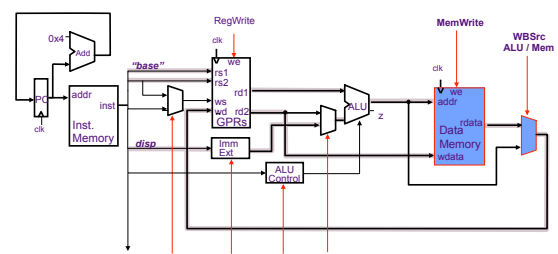
- single read/write memory for program and data

- Note:
A Load or Store instruction requires accessing the memory more than once during its execution

CSE 490/590, Spring 2011

23

Load/Store Instructions: Harvard Datapath



CSE 490/590, Spring 2011

24

Acknowledgements

- These slides heavily contain material developed and copyright by
 - Krste Asanovic (MIT/UCB)
 - David Patterson (UCB)
- And also by:
 - Arvind (MIT)
 - Joel Emer (Intel/MIT)
 - James Hoe (CMU)
 - John Kubiatowicz (UCB)
- MIT material derived from course 6.823
- UCB material derived from course CS252