Programmable Logic Array (PLA)

In a ROM, the AND section is a decoder that generates all the 2n outputs. The OP section can be programmed according to our design needs.

In a PLA, both the AND section and the OR section can be programmed. By programming the AND section, we generate only those boolean product terms that we need.

Any combinational circuit can be designed using a PLA. Any sequential circuit (therefore, any control unit) can be designed using a set of flip-flops and a PLA. How?

Design of the MIPS processor

We will study the design of a simple version of MIPS that can support the instructions

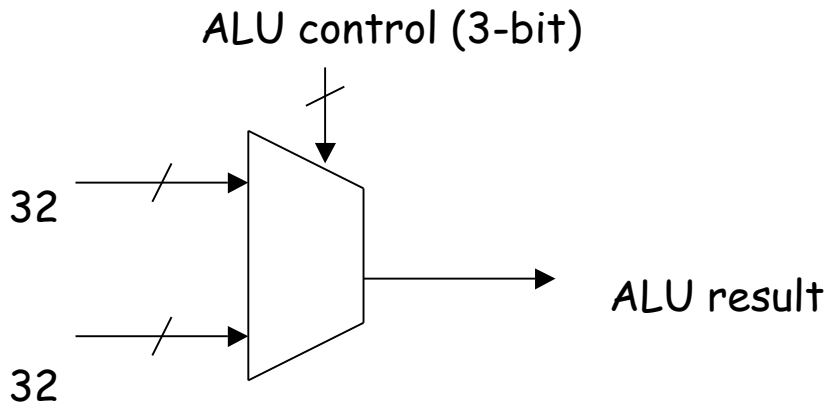
- I-type instructions LW, SW
- R-type instructions, like ADD, SUB
- Conditional branch instruction BEQ
- J-type branch instruction J

We will closely follow the material from Chapter 5 of the textbook. Follow the class lectures.

The instruction formats

	6-bit	5-bit	5-bit	5-bit	5-bit	5-bit
LW	op	rs	rt	immediate		
SW	op	rs	rt	immedeiate		
ADD	op	rs	rt	rd	0	func
SUB	op	rs	rt	rd	0	func
BEQ	op	rs	rt	immediate		
J	op	address				

ALU control



ALU control input	ALU function
000	AND
001	OR
010	add
110	sub
111	Set less than

How to generate the ALU control input? The control unit first generates a **2-bit ALU op** from the opcode of the instruction. This information is combined with the **function field** of some of the R-type operations to generate the **3-bit control input**.

Design of MIPS (continued)

Follow the design steps from the class lectures.

We will study the datapath, the control unit, and the performance of the simple version of MIPS that executes **every instruction in one cycle**. Read these from Chapter 5.

Observe the role of the control signals.

After this, we will study the more realistic version of MIPS, the multi-cycle version.

The following diagrams are relevant. Study them carefully: Fig 5.9, 5.13 – 5.19, 5.29. Also see Fig, C.1 – C.4

Multi-cycle Implementation of MIPS

Why is the single-cycle implementation of MIPS inefficient?

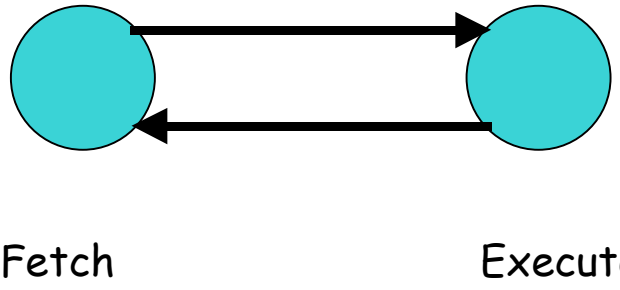
The cycle-time of the 1 CPI, Cycles Per Instruction MIPS will be equal to the time required to complete the longest instruction. This will slow down the execution. This implementation, although quite simple, is not attractive.

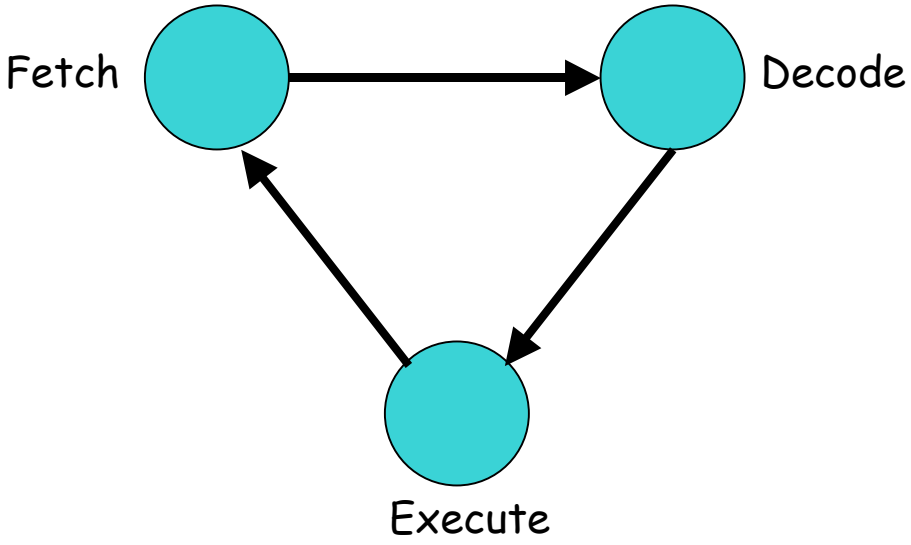
(See figure 5.30 for a basic multi-cycle datapath. This diagram combines the instruction memory and the data memory into one unit, although this is not essential. In fact most designs these days have separate instruction and data memory)

In the multi-cycle implementation, each instruction takes a few cycles. The number of cycles is different for different instructions. Note the various buffers between the stages.

The Instruction Cycle

The instruction cycle tells us what the processor does in each clock cycle.



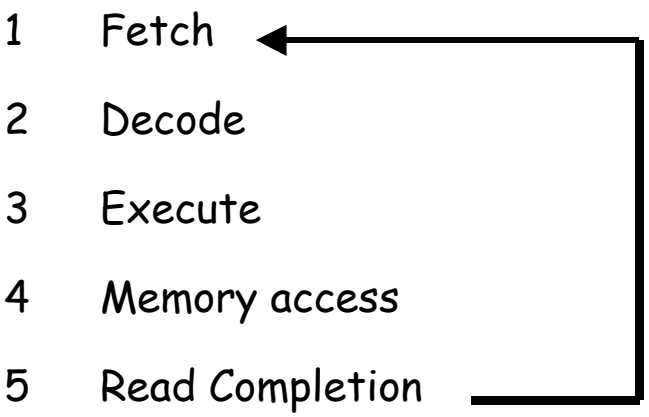


Pipelining

Study Chapter 6.1. Understand how instruction pipelining speeds up computation. What are the bottlenecks in a pipeline?

Instruction Cycle for Multi-cycle MIPS

(See Fig. 5.31, 5.32, 5.33 and 5.34, 5.36, 5.37, 5.42)



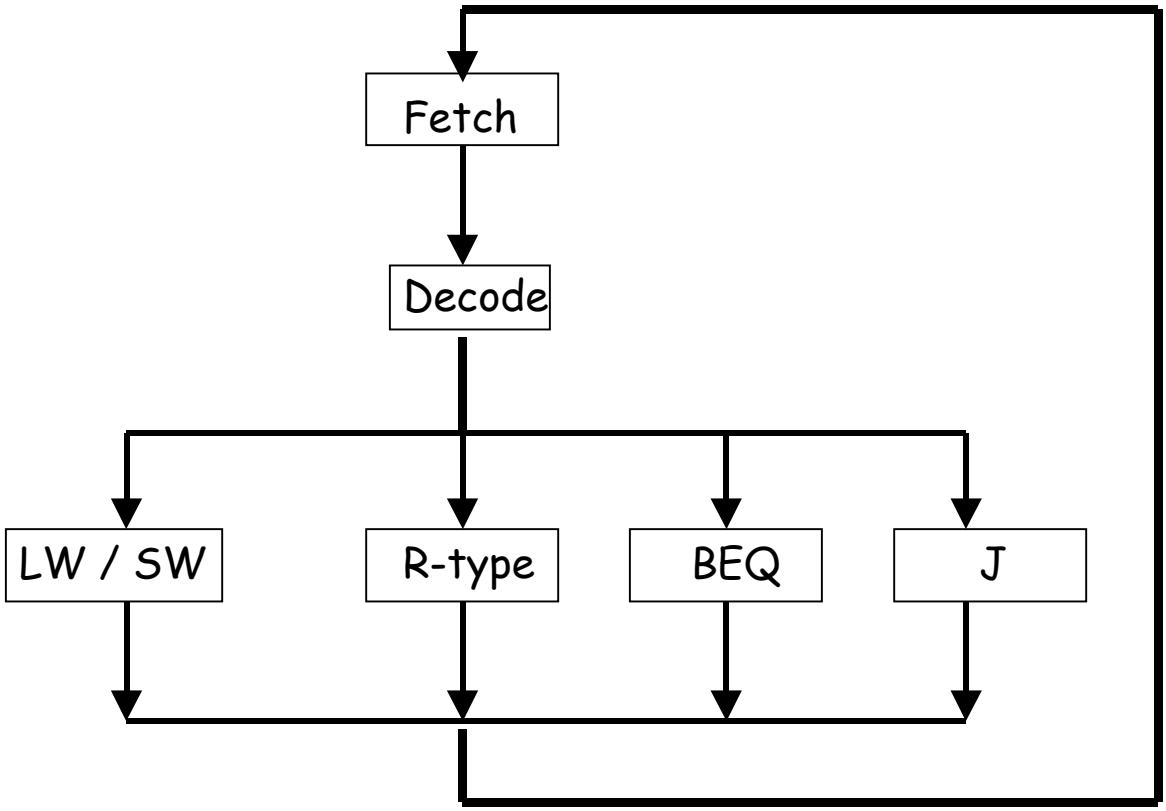
The first two steps are the same for all instructions. The last three steps are different for different instructions. Fig. 5.35 contains a summary.

Understanding the Control Unit

1 Hardware Control (also called Finite State Control)

2 Micro-programmed Control

Finite State Control



Study Fig. C.6 – C.7

Memory Hierarchy

Cache Memory (Chapter 7.1-7.3)

Disk and disk characteristics (Part of Chapter 8.3)

Virtual Memory (Chapter 7.4)

Input Output Devices

Buses (Read Chapter 8.4)

Shared Communication Link consisting of data and control lines.

Processor-to-Memory and I/O buses (See Fig. 8.9)

What is a bus transaction?

How to increase the bus bandwidth? Use wider data bus.

Bus access and bus arbitration – Who is a bus master?

Can there be multiple bus masters? How to avoid the chaos?

Device Addressing

Memory mapped devices vs. dedicated I/O instruction

Polled I/O (Chapter 8.5)

Interrupt driven I/O, Device polling vs. daisy chaining

Direct Memory Access (DMA)