Topic for today:

More on the control unit: microprogramming

Recall: Decoding opcodes

Questions:

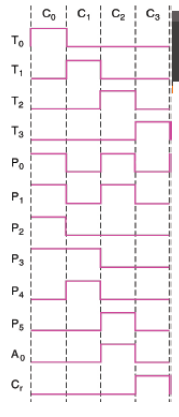
How do you get the settings needed for a register transfer from a single opcode?

How can you get a sequence of transfers from a single opcode?

Recall: Hardwired control

In a hardwired approach, the opcode is decoded by a large set of digital circuits (decoders, multiplexers, etc.) to send the correct signals to the registers, ALU, and internal CPU bus controller. (This is all just Boolean algebra!)

Also, the clock signal is connected to a counting circuit to ensure that actions happen in the correct sequence.



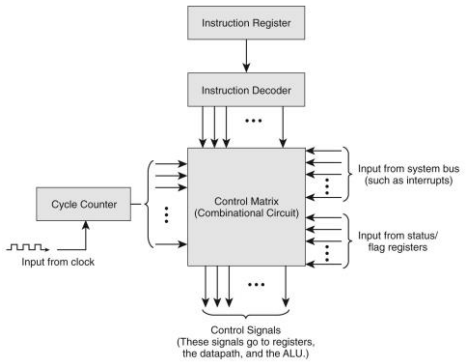
ADD Instruction Control

- Add instruction timing diagram
- Notice the concurrent signal states during each machine cycle: C0 through C3.

```
P3 P2 P1 P0 T0: MAR ← X
      P4 P3 T1: MBR ← M[MAR]
A0 P5 P1 P0 T2: AC ← AC + MBR
         Cr T3: [Reset counter]
```

4

Hardwired Control Unit



5

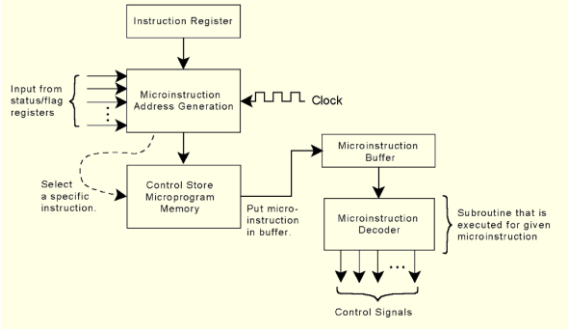
Microprogrammed control

In a microprogramming approach, the program that is "really" running is a relatively short program, stored in the CPU itself (*not* in main memory), whose function is to run the fetch-decode-execute cycle. It fetches an instruction of the macroprogram from memory, decodes its opcode, carries out the specified operation, and repeats. Each of these steps requires a sequence register transfers, which take one clock cycle each.

Microprogram

The microprogram is stored in a part of the control unit called the control store. It is some version of read-only memory (ROM).

Microprogrammed control unit



8

Tradeoffs

A microprogrammed processor is

- easier to design
- easier to debug
- easier to modify.

A hardwired processor is faster.

Historical note on microprogramming

- Proposed by M. V. Wilkes in 1951
- Idea was ignored in practice until the early 1960s – IBM 360 family
- Less common today

Real world Architectures

- MARIE shares many features with modern architectures but it is not an accurate depiction of them.
- In the following slides, we briefly examine two machine architectures.
- We will look at an Intel architecture, which is a CISC machine and MIPS, which is a RISC machine.
  - CISC is an acronym for complex instruction set computer.
  - RISC stands for reduced instruction set computer.

Real world Architectures

- Each member of the x86 family of Intel architectures is known as a CISC (Complex Instruction Set Computer) machine, whereas the Pentium family and the MIPS architectures are examples of RISC (Reduced Instruction Set Computer) machines.
- The main objective of RISC machines is to simplify instructions so they can execute more quickly. Each instruction performs only one operation; they are all the same size.

Real world Architectures

- MIPS- a reduced instruction set computer (RISC) instruction set (ISA) developed by MIPS Technologies
- If you like MARIE and want to try a more complex machine, you should try the MIPS Assembler and Runtime Simulator (MARS)
- MARS is a Java-based MIPS simulator