COMPUTER ORGANIZATION AND DESIGN
The Hardware/Software Interface
5th Edition

Appendix D

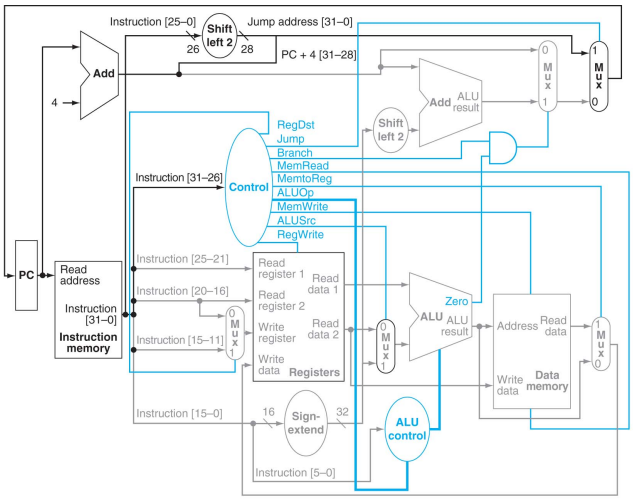
Controller Implementation

Controller Implementations

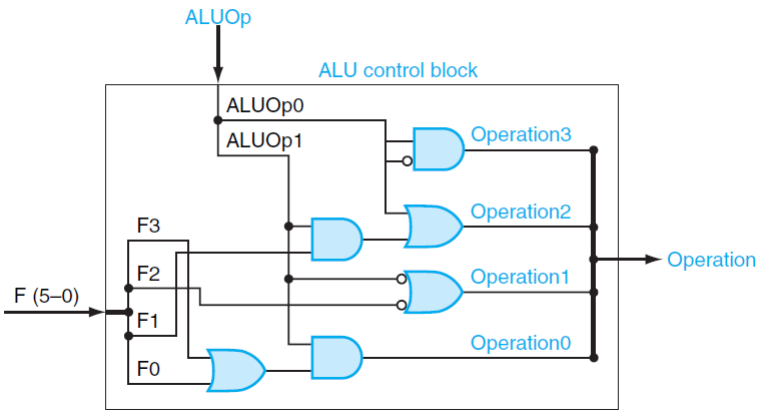
- Combinational logic (single-cycle);
- Finite state machine (multi-cycle, pipelined);
- ROM;
- PLA;
- Micro-code.

Single Cycle Implementation

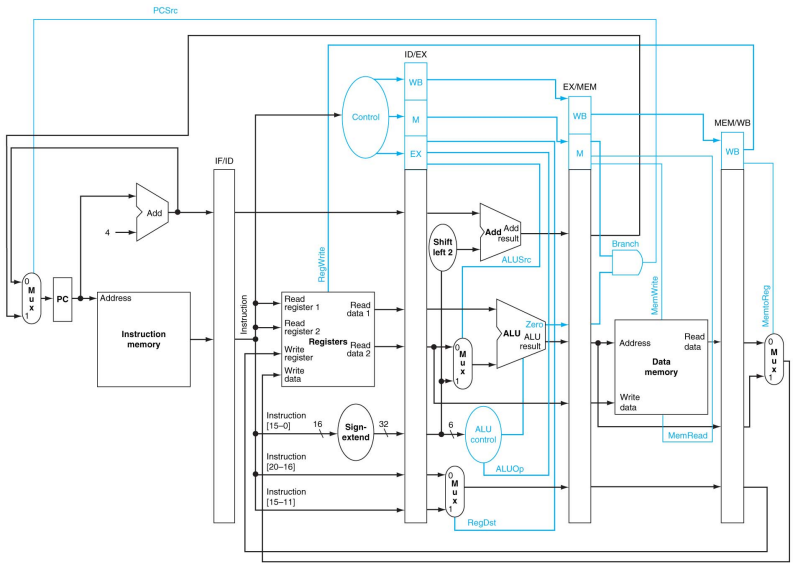
- All instructions take one clock cycle.
- Clock rate determined by the slowest instruction (LW).



Combinational Implementation



Pipelined Implementation

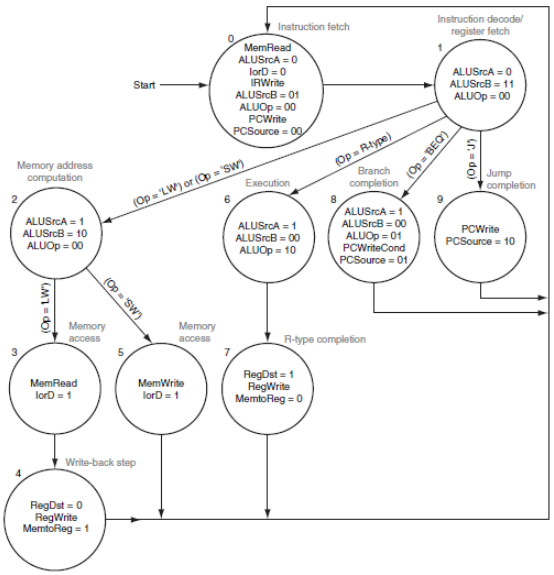


Implementing Control

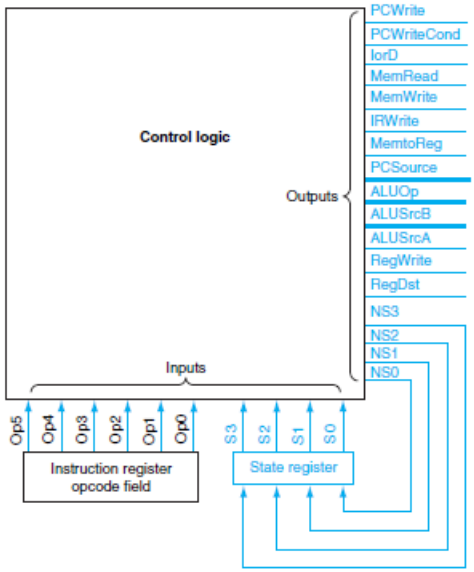
- Value of control signals is dependent upon
  - What instruction is being executed.
  - Which step is being performed.
- Use the information we've accumulated to specify a finite state machine
  - Specify the finite state machine graphically.
  - Use micro-programming.

Graphical Specification of FSM

- Control signals
  - Are “don’t care” if they are not mentioned.
  - Are asserted if name only.
  - Otherwise the value is stated.
- How many state bits will we need?

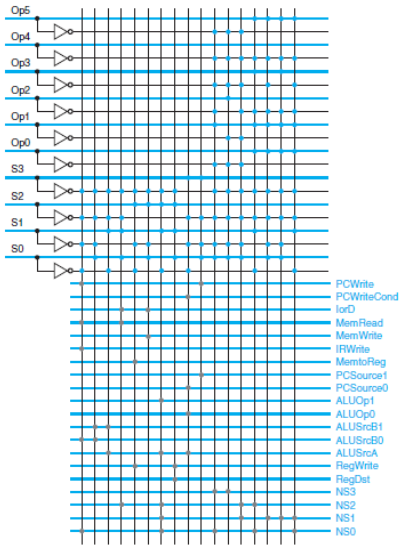


Finite State Machine for Control



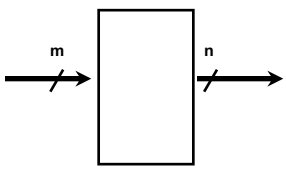
PLA Implementation

- If I picked a horizontal or vertical line, could you explain it?



ROM Implementation

- ROM = "Read Only Memory"
  - Values of memory locations are fixed ahead of time.
- A ROM can be used to implement a truth table
  - If the address is m-bits, we can address 2^m entries in the ROM.
  - Outputs are the bits of data that the address points to.
  - *m* is the "height", and *n* is the "width."



0	0	0	0	0	1	1
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	0	1	1	0
1	1	1	0	1	1	1

ROM Implementation

- How many inputs are there?
  - 6 bits for opcode, 4 bits for state = 10 address lines.
  - i.e. 2^{10} = 1024 different addresses.
- How many outputs are there?
  - 16 datapath-control, 4 state bits = 20 outputs.
- ROM is 2^{10} x 20 bits = 20K bits (a rather unusual size).
- Rather wasteful, since for lots of entries the outputs are the same.

ROM vs. PLA

- PLA is much smaller
  - Can share product terms.
  - Only need entries that produce an active output.
  - Can take into account don't cares.
- Size is (#inputs × #product-terms) + (#outputs × #product-terms).
  - For this example = (10x17)+(20x17) = 510 PLA cells
- PLA cell is slightly larger than a ROM cell.

Micro-programming Prelude

- The controller is easy to graphically specify for the few instructions we are implementing.
- What about a full MIPS instruction set with over 100 instructions ranging from 1 clock cycle to over 20 clock cycles?
  - Use VHDL or Verilog, but inefficient from hardware perspective.
- Consider an instruction set with several hundred instructions of widely varying classes, such as the IA-32 architecture
  - Control unit could easily require thousands of states with hundreds of different sequences.
  - Specifying the control unit with a graphical representation would be impossible.

Micro-programming

- Suppose we think of the set of control signals that must be asserted in a state as an instruction to be executed by the datapath. To avoid confusing the instructions of the MIPS instruction set with these low-level control instructions, the latter are called ***micro-instructions***.
- Each micro-instruction defines the set of datapath control signals that must be asserted in a given state.
- Executing a micro-instruction has the effect of asserting the control signals specified by the micro-instruction.

Micro-programming

- In addition to defining which control signals must be asserted, we must also specify the sequencing—what micro-instruction should be executed next?
- If the micro-instruction requirements become large, than a “micro-instruction assembler” is usually used, including such abilities as subroutine calls.
- Designing the control as a program that implements the machine instructions in terms of simpler micro-instructions is called *micro-programming.*

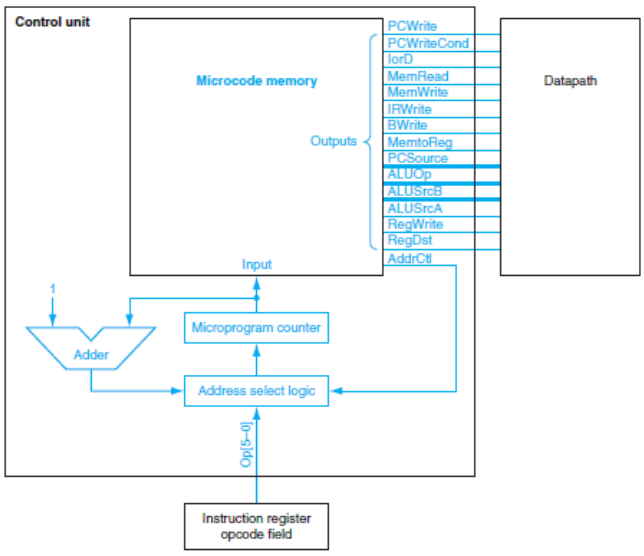
Micro-programming aka Wikipedia

- Microcode is stored in SRAM or flash memory. This is traditionally denoted a "writeable control store" in the context of computers. Complex digital processors may also employ more than one (possibly microcode based) control unit in order to delegate sub-tasks which must be performed (more or less) asynchronously in parallel. Microcode is generally not visible or changeable by a normal programmer, not even by an assembly programmer. Unlike machine code which often retains some compatibility among different processors in a family, microcode only runs on the exact electronic circuitry for which it is designed.

Micro-programming aka Wikipedia

- More extensive micro-coding has also been used to allow small and simple microarchitectures to emulate more powerful architectures with wider word length, more execution units and so on; a relatively simple way to achieve software compatibility between different products in a processor family.
- Some hardware vendors, especially IBM, use the term as a synonym for *firmware*, so that all code in a device, whether microcode or machine code, is termed microcode (such as in a hard drive for instance, which typically contains both).

Micro-program Controller



Maximally vs. Minimally Encoded

- No encoding
  - Basis for VLIW.
  - 1 bit for each data path control signal.
  - Faster, requires more memory.
  - Used for Vax 780 in the 1980's — 400K of memory.
- Lots of encoding
  - Send the micro-instructions through logic to get control signals.
  - Uses less memory, slower.

Vocabulary

- Micro-instruction
  - Contains a control word and a sequencing word
    - Control Word - all the control information required for one clock cycle.
    - Sequencing Word - information needed to decide the next micro-instruction to be executed.
- Control Memory or Control Storage
  - Writable storage in the micro-programmed control unit to store the micro-program.
  - Allows the micro-program to be modified, thereby providing means to change or modify the instruction set.

Micro-instruction Classification

- Micro-instructions can be classified in a variety of ways in which the designer must choose the parallel “power” of each instruction. There are two main types:
  - Vertical micro-programming - each micro-instruction specifies a single (or few) micro-operations to be performed.
  - Horizontal micro-programming - each micro-instruction specifies many different micro-operations to be performed in parallel.

Micro-programming

- Vertical
  - Width is narrow - *N* control signals can be encoded into *log2n* control bits.
  - Limited ability to express parallelism.
  - Considerable encoding of control information requires external memory word decoder to identify the exact control lines being manipulated.
- Horizontal
  - Wide memory word.
  - High degree of parallel operations are possible.
  - Little to no encoding of control information.

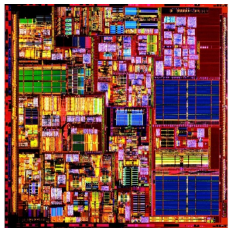
Compromise Technique

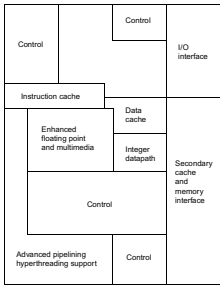
- Nano-programming
  - Use a 2-level control storage organization.
  - Top level is a vertical format memory.
  - Output of the top level memory drives the address register of the bottom (nano-level) memory.
  - Nano-memory uses the horizontal format which produces the actual control signal outputs.
  - Main advantage is significant saving in control memory size.
  - Main disadvantage is more complexity and slower operation (doing 2 memory accesses for each micro-instruction).

Historical Perspective

- In the ‘60s and ‘70s micro-programming was used frequently for implementation.
- This led to more sophisticated ISAs and the VAX.
- In the ‘80s RISC processors based on pipelining became popular.
- Pipelining the micro-instructions is also possible.
- Implementations of IA-32 processors since the 486 use
  - “hardwired control” for simpler instructions (few cycles, FSM control implemented using PLA or random logic).
  - “micro coded control” for more complex instructions (large numbers of cycles).

Pentium 4





- Processor executes simple microinstructions, 70 bits wide (hardwired).
- 120 control lines for integer datapath, 400 for floating point.
- If an instruction requires more than 4 microinstructions to implement, control from micro-code ROM (8000 micro-instructions).
- It is complicated!

Summary

- Finite-state -machines give the most flexibility
  - PLA
  - ROM
  - Micro-programming
- Modern processors are complicated
  - Micro-coded.
  - Make the common case fast.
  - Make the simple instructions fast.
  - Take the performance hits on the complex instructions.