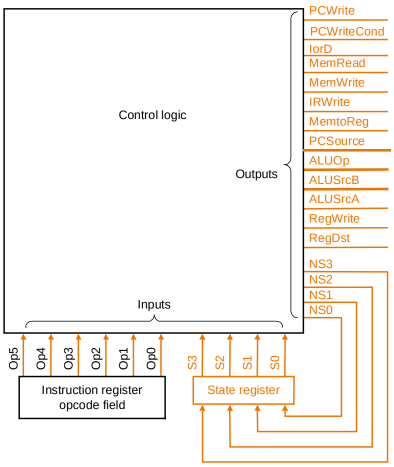
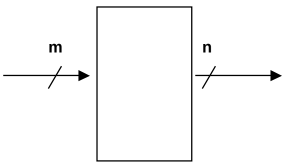
Finite State Machine for Control

Implementation



ROM Implementation

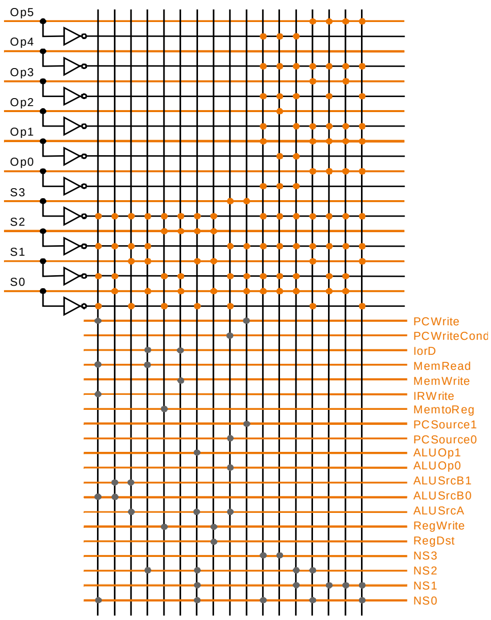
- ROM = "Read Only Memory"
  - values of memory locations are fixed ahead of time
- A ROM can be used to implement a truth table
  - if the address is m-bits, we can address 2^m entries in the ROM.
  - our outputs are the bits of data that the address points to.



ROM Implementation (cont.)

- 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 outputs, 4 state bits = 20 outputs
- ROM is $2^{10} \times 20$ = 20K bits (and a rather unusual size)
- Rather wasteful, since for lots of the entries, the outputs are the same
  i.e., opcode is often ignored, same contents are duplicated 64 times.
- Break up the table into two parts
  4 state bits tell you the 16 outputs, $2^4 \times 16$ = 0.3 Kbits of ROM
  10 bits tell you the 4 next state bits, $2^{10} \times 4$ = 4 Kbits of ROM
  Total: 4.3K bits of ROM

PLA Implementation

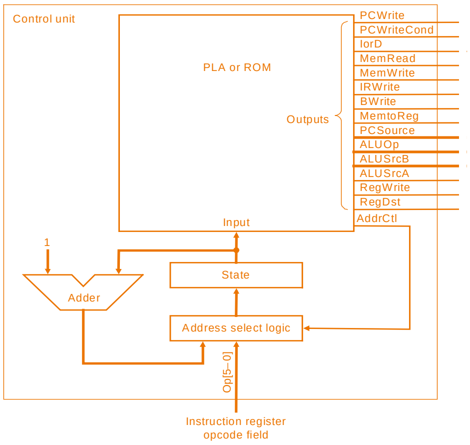


PLA Implementation (cont.)

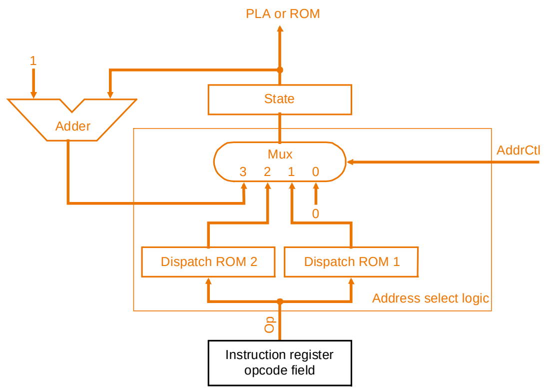
- PLA is much smaller
  - can share product terms
  - only need entries that produce an active output
  - can take into account don't cares
- Size is (#inputs × #product-terms) + (#outputs × #product-terms)
  - For this example = (10 × 17)+(20 × 17) = 460 PLA cells
- PLA cells usually about the size of a ROM cell (slightly bigger)

Simplifying Next State Function

Complex instructions: the "next state" is often current state + 1



Details



Details (cont.)

Dispatch ROM 1		
Op	**Opcode name**	**Value**
000000	R-format	0110
000010	jmp	1001
000100	beq	1000
100011	lw	0010
101011	sw	0010

Dispatch ROM 2		
Op	**Opcode name**	**Value**
100011	lw	0011
101011	sw	0101

State number	Address-control action	Value of AddrCtl
0	Use incremented state	3
1	Use dispatch ROM 1	1
2	Use dispatch ROM 2	2
3	Use incremented state	3
4	Replace state number by 0	0
5	Replace state number by 0	0
6	Use incremented state	3
7	Replace state number by 0	0
8	Replace state number by 0	0
9	Replace state number by 0	0

CPI of Multi-cycle Machine

Assume instruction mix are the following:

Instruction	Clock cycles	frequency
Load	5	22
Store	4	11
R-format	4	49
Branch	3	16
Jump	3	2

$$AverageCPI = \sum (CPI_i \times f_i)$$
$$= (5\times 24\% + 4\times 12\% + 4\times 44\% + 3\times 18\% + 3\times 2\%)$$
$$= 4.04$$

If all instructions take the same # of clock cycles, CPI will be 5.