

FIGURE 5.39 The multicycle datapath with the addition needed to implement exceptions. The specific additions include the Cause and EPC registers, a multiplexor to control the value sent to the Cause register, an expansion of the multiplexor controlling the value written into the PC, and control lines for the added multiplexor and registers. For simplicity, this figure does not show the ALU overflow signal, which would need to be stored in a one-bit register and delivered as an additional input to the control unit (see Figure 5.40 to see how it is used).

Figure 5.40 represents a complete specification of the control for this MIPS subset with two types of exceptions. Remember that the challenge in designing the control of a real machine is to handle the variety of different interactions between instructions and other exception-causing events in such a way that the control logic remains both small and fast. The complex interactions that are possible are what make the control unit the most challenging aspect of hardware design.

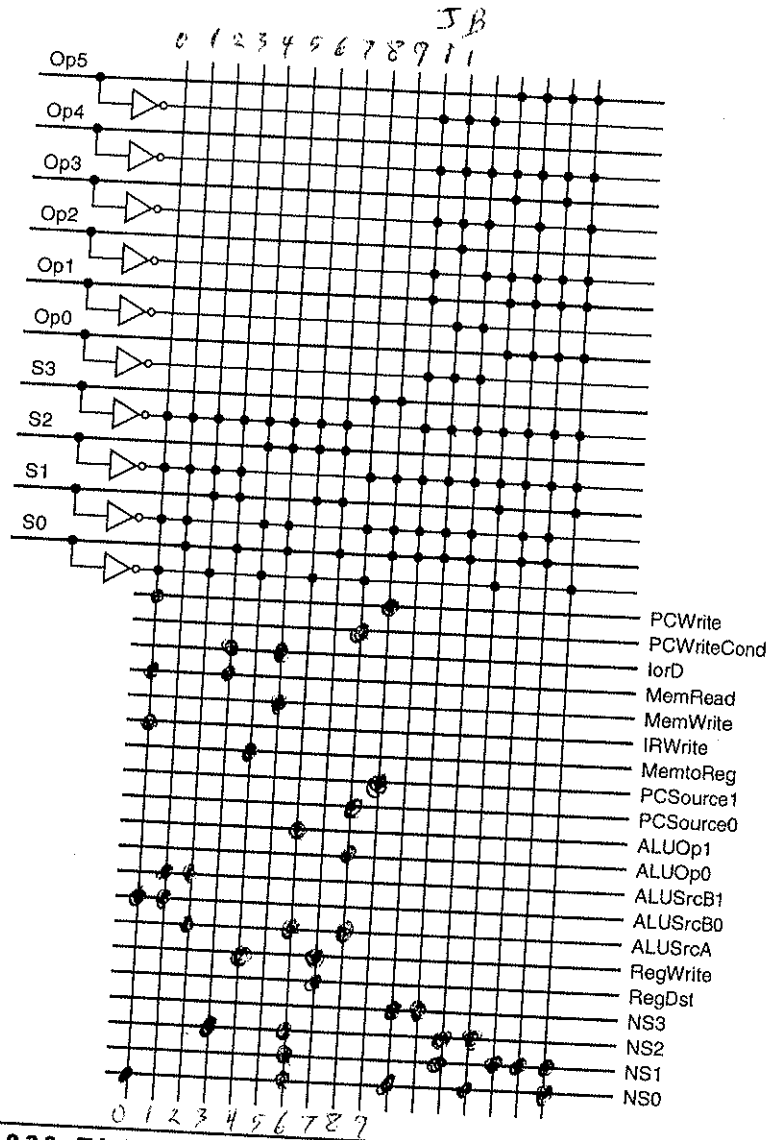


FIGURE C.3.9 This PLA implements the control function logic for the multicycle implementation. The inputs to the control appear on the left and the outputs on the right. The top half of the figure is the AND plane that computes all the minterms. The minterms are carried to the OR plane on the vertical lines. Each colored dot corresponds to a signal that makes up the minterm carried on that line. The sum terms are computed from these minterms, with each grey dot representing the presence of the intersecting minterm in that sum term. Each output consists of a single sum term.

Appendix C Mapping Control to Hardware

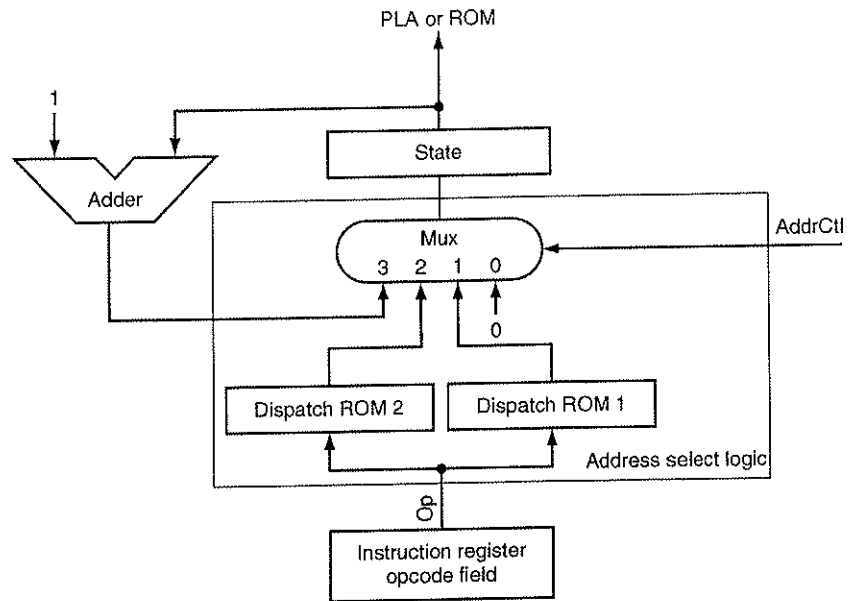


FIGURE C.4.2 This is the address select logic for the control unit of Figure C.4.1.

the instruction set, so the dispatch ROMs will be largely empty. Figure C.4.3 shows the entries that must be assigned for this subset. Section 5.6 of Chapter 5 discusses what to do with the entries in the dispatch ROMs that do not correspond to any instruction.

Dispatch ROM 1			Dispatch ROM 2		
Op	Opcode name	Value	Op	Opcode name	Value
000000	R-format	0110	100011	lw	0011
000010	jmp	1001	101011	sw	0101
000100	beq	1000			
100011	lw	0010			
101011	sw	0010			

FIGURE C.4.3 The dispatch ROMs each have $2^8 = 64$ entries that are 4 bits wide, since that is the number of bits in the state encoding. This figure only shows the entries in the ROM that are of interest for this subset. The first column in each table indicates the value of Op, which is the address used to access the dispatch ROM. The second column shows the symbolic name of the opcode. The third column indicates the value at that address in the ROM.

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

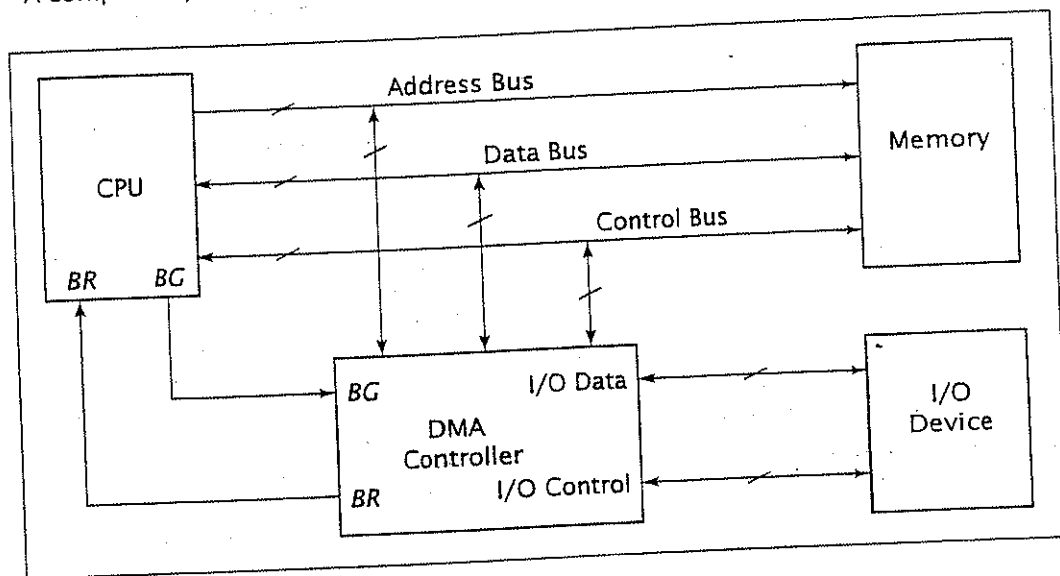
FIGURE C.4.4 The values of the address-control lines are set in the control word that corresponds to each state.

State number	Control word bits 17-2	Control word bits 1-0
0	1001010000001000	11
1	0000000000011000	01
2	0000000000010100	10
3	0011000000000000	11
4	0000001000000010	00
5	0010100000000000	00
6	0000000001000100	11
7	0000000000000011	00
8	0100000010100100	00
9	100000001000000000	00

FIGURE C.4.5 The contents of the control memory for an implementation using an explicit counter. The first column shows the state, while the second shows the datapath control bits, and the last column shows the address-control bits in each control word. Bits 17-2 are identical to those in Figure C.3.7.

nections. In this configuration, the DMA controller is sometimes referred to as a **channel**. In an alternate configuration, the DMA controller may be incorporated directly into the I/O device.

Figure 10.17
A computer system with DMA



To transfer data from an I/O device to memory via DMA, internal configuration of a DMA controller. All buffer enable signals (not shown) are supplied by the DMA control unit and all register load signals (not shown) are derived from the system address and control buses.

