

Homework #7
Due Wed. 11/30

1. (H&J 5.3)

Use a sheet similar to Figure 5.7 of the text to trace the code fragment given on page 191 through the pipeline. Assume the following decimal register and memory values. $r1=1$, $r2=-4$, $r3=32$, $r5=8$, $addr1=1000$, $addr2=2000$, $PC=100$, $M[2000]=100$. Insert `nop` bubbles into the pipeline as needed to resolve any dependencies.

Code:

```
shr  r3, r3, 2    ;Storing result into r3
sub  r2, r5, r1   ;Idle - no memory access needed
add  r4, r3, r2   ;Performing addition in ALU
st   r4, addr1   ;Accessing r4 and addr1
ld   r,  addr2    ;Fetching instruction
```

2. (H&J 5.9)

For the following pairs of instructions,

- (a) indicate how many bubbles must be placed between them in the presence of and in the absence of data forwarding to resolve the dependence.
- (b) trace each pair through a sheet similar to Figure 5.7. (For each, just indicate the the stage location for the instructions and the registers involved with the data forwarding)

1. <code>la r2, (r4)</code> <code>shc r6, r4, 42</code>	2. <code>not r2, r4</code> <code>sub r6, r2, r0</code>	3. <code>lar r31, -12</code> <code>brl r31, r30</code>
4. <code>add r2, r0, r4</code> <code>st r0, 12(r2)</code>	5. <code>add r2, r2, r4</code> <code>st r2, 12(r2)</code>	6. <code>brl r31, r30</code> <code>shl r31, r30, 2</code>

3. (H&J 5.12a)

The RTN fragments 5.1 - 5.3 on pages 215 and 216 describe the hazard detection with bubble insertion for the 2-operand ALU-ALU instructions.

- (a) Write analogous equations to detect hazards between an `alu` instruction followed by a `ld` instruction

4. (H&J 5.14)

Design the digital logic circuitry of the 2-operand ALU-ALU portion of the stage 4 hazard detection and forwarding unit shown in Figure 5.15. (You can ignore the Z5 signal when determining ALU inputs. You may want to remember the control signals defined in Fig. 5.3.)

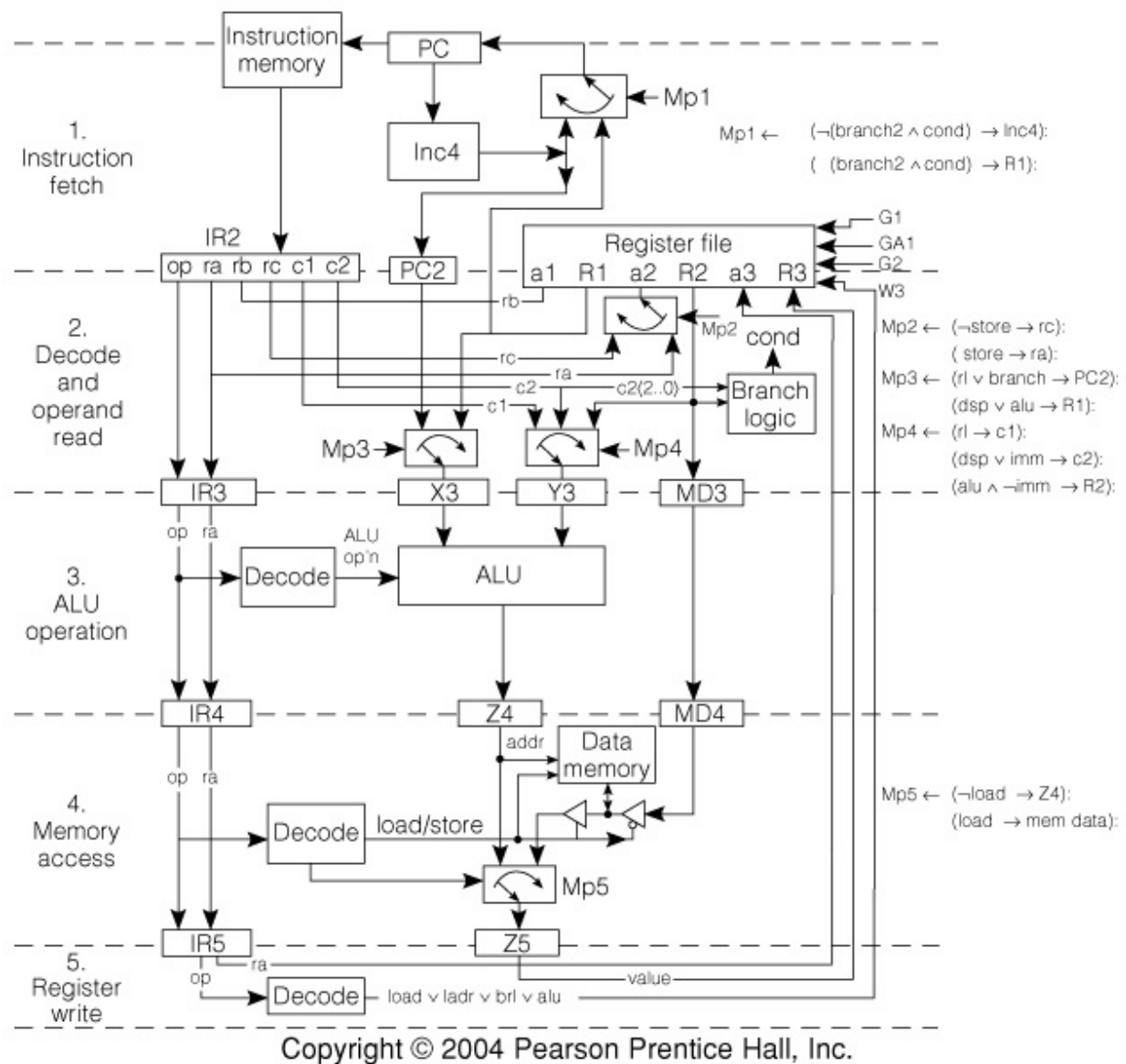


Figure 5.7: The Pipeline Data Path with Selected Control Signals

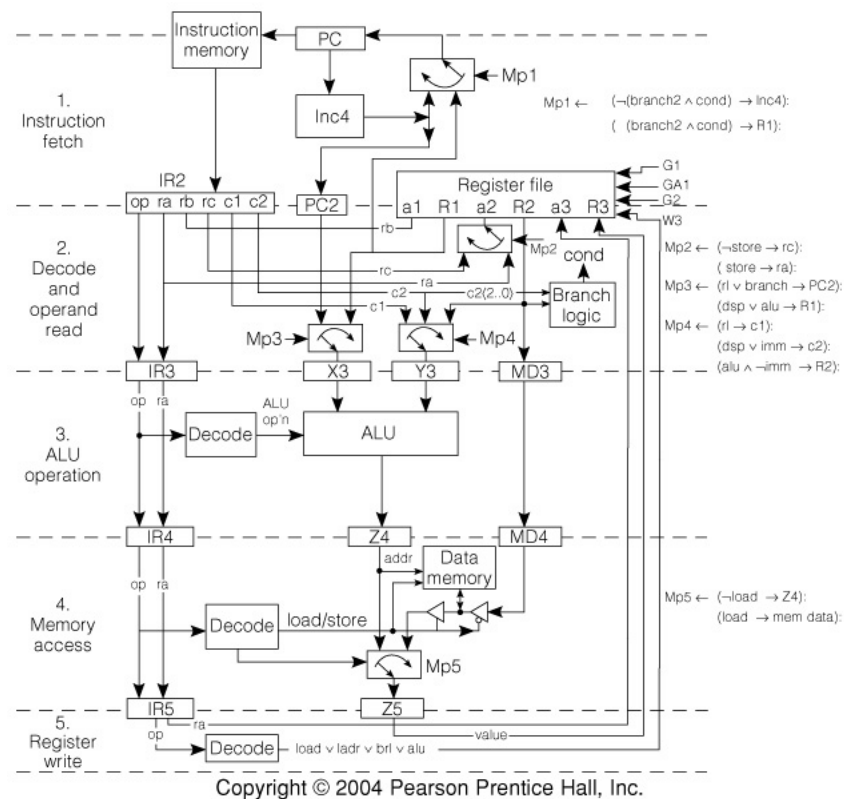


Figure 5.7: The Pipeline Data Path with Selected Control Signals