

- 4.2** Extend the SRC instruction set by adding the instruction swap ra, rb, (op = 7), that exchanges the contents of the two specified registers.
- Define a plausible abstract RTN for this instruction.
 - Develop the concrete RTN for it, assuming the 1-bus SRC microarchitecture. (§4.4)

Solution: a. swap(:= op = 7) → (R[ra] ← R[rb]; R[rb] ← R[ra]);

b.	Step	RTN
	T0.	MA ← PC; C ← PC + 4;
	T1.	MD ← M[MA]; PC ← C;
	T2.	IR ← MD;
	T3.	MD ← R[ra];
	T4.	C ← R[rb];
	T5.	R[ra] ← C;
	T6.	R[rb] ← MDS;

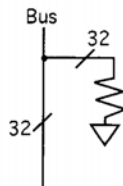
4.7

The SRC instruction set includes the neg instruction, which computes the arithmetic 2's complement negation of a register operand. Assume that the NEG operation is not in the set of operations the ALU can perform. (See page 157.) Develop the concrete RTN and the control sequence to implement the neg instruction for the 1-bus design. (§4.4)

Solution: The NEG instruction can be implemented using the SUB instruction by subtracting the operand from zero to get its negative. Concrete RTN and control sequences for instruction NEG for 1-bus design follow:

Step	RTN	Control sequence
T0.	MA ← PC; C ← PC + 4;	PC _{out} , MA _{in} , INC4, C _{in}
T1.	MD ← M[MA]; PC ← C;	C _{out} , PC _{in} , Read, Wait
T2.	IR ← MD;	MD _{out} , IR _{in}
T3.	A ← 0 [†] ;	A _{in}
T4.	C ← A - R[rc];	Gr _c , R _{out} , SUB, C _{in}
T5.	R[ra] ← C;	C _{out} , Gra, R _{out} , End

[†]Note: Assume a passive pull-down on the bus, so that it is 0 if no 3-state gates are active.



4.12

In the section on performance estimation on page 173 it was arbitrarily assumed that the minimum clock period of the 2-bus design would be 1.1 times the clock period of the 1-bus design. Given the delays in Exercise 4.8, revisit this problem.

- a. Calculate the minimum clock period for the 1-bus and 2-bus designs, assuming a 25% increase in clock periods as a safety factor.
- b. Using the clock periods you calculated, compute the speedup to be expected for the 2-bus design. (§4.6)

Solution: a. Maximum total delay of 1-bus
 = control unit + reg file (Fig 4.4) + bus prop + CON ckt (Fig 4.9)
 = 10 + (6 gates in reg file) + (7 ns bus prop) + (4 gates + 1 latch in CON ckt)
 = 10 + 6·3 + 7 + 4·3 + 6 = 53 ns

Clock period of 1-bus = Maximum total delay of 1-bus × (1 + 0.25) = 66.25 ns

Maximum total delay of 2-bus
 = Maximum total delay of 1-bus + bus prop = 53 + 7 = 60 ns

Clock period of 2-bus = Maximum total delay of 2-bus × (1 + 0.25) = 75 ns

b. Here we make the same assumption as in Section 4.6.1, that is, all instructions of the 1-bus design will execute in 8 clock cycles and all instructions of the 2-bus design will execute in 7 clock cycles.

% Speedup =

$$\frac{IC \times 8 \times \tau_{1\text{-bus}} - IC \times 7 \times \tau_{2\text{-bus}}}{IC \times 7 \times \tau_{2\text{-bus}}} \times 100 = \frac{8 \times 66.25 - 7 \times 75}{7 \times 75} \times 100$$

= 0.95%

4.19

```
reset (:= op = 18) → (
  c1<0> = 0 → PC, R[0..31] ← 0:
  c1<0> = 1 → PC ← 0;
  instruction_execution );
```