Problem 4:

1. Sign-extend = 00000000000000000000000000010100
   Jump shift-left-2 = 0001100010000000000001010000

2. ALUop = 00
   Instruction = 010100

3. NewPC = PC+4
   Path = PC to ADD to branch-mux to jump-mux to PC

4. WrReg-mux = 2 or 0
   ALU-mux = 20
   Mem/ALU-mux = X
   Branch-mux = PC+4
   Jump-mux = PC+4

5. ALU = -3 and 20
   Add (PC+4) = PC and 4
   Add (branch) = PC+4 and 20*4

6. ReadReg 1 = 3
   ReadReg 2 = 2
   WriteReg = 2 or 0
   WriteData = X
   RegWrite = 0

Problem 5:

1. Pipelined = 350ps
   Single-Cycle = 1250 ps

2. Pipelined = 1750ps
   Single-Cycle = 1250

3. Stage to split = ID
   New Cycle Time = 300ps

4. 35%

5. 65%

6. Multi-cycle execution time is 4.2 times pipelined execution
   Single-cycle execution time is 3.57 times pipelined execution

Problem 6:

1. RAW on R1 from I1 to I2 and I3
   RAW on R2 from I2 to I3
   WAR on R2 from I1 to I2
   WAR on R1 from I2 to I3
   WAW on R1 from I1 to I3
2. Add 2 NOPs between each instruction for RAW hazards on R1 and R2
3. No hazards
4. No forwarding = 1980ps
   Forwarding = 1680ps
   Speedup = 1.18
5. No NOPs needed
6. No forwarding = 1980ps
   ALU-ALU forwarding = 1470ps
   Speedup = 1.35

Problem 7:

1. RegWrite = 1
   MemRead = 0
   ALUMux = 0
   MemWrite = 0
   ALUop = AND
   RegMux = 0
   Branch = 0
2. All except data memory, immediate sign-extender, and branch adder
3. Not used = branch adder
   No output = data memory

Problem 8:

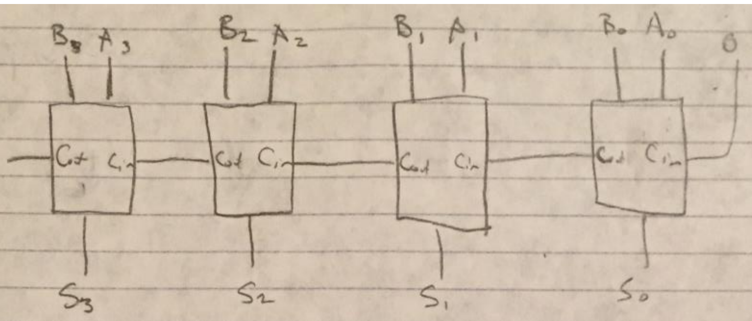
Easiest = “bit equal” – would only need an optional negation after the xor

Medium = “replace under mask” – would require new function for ALU

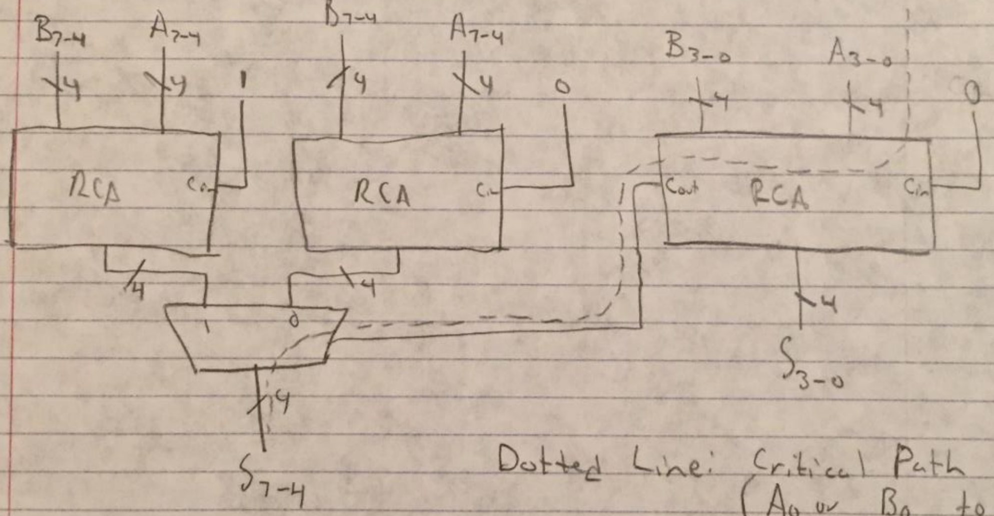
Hard = “split register” – This requires writing two registers which would require an additional data path through many of the stages and an additional write port for the register file

Problem 9:

Ripple-Carry



Carry-Select:



Dotted Line: Critical Path
(A0 or B0 to S7-4)

Delay = 10 x nand2 + 1X NOT
= 129T

Problem 10:

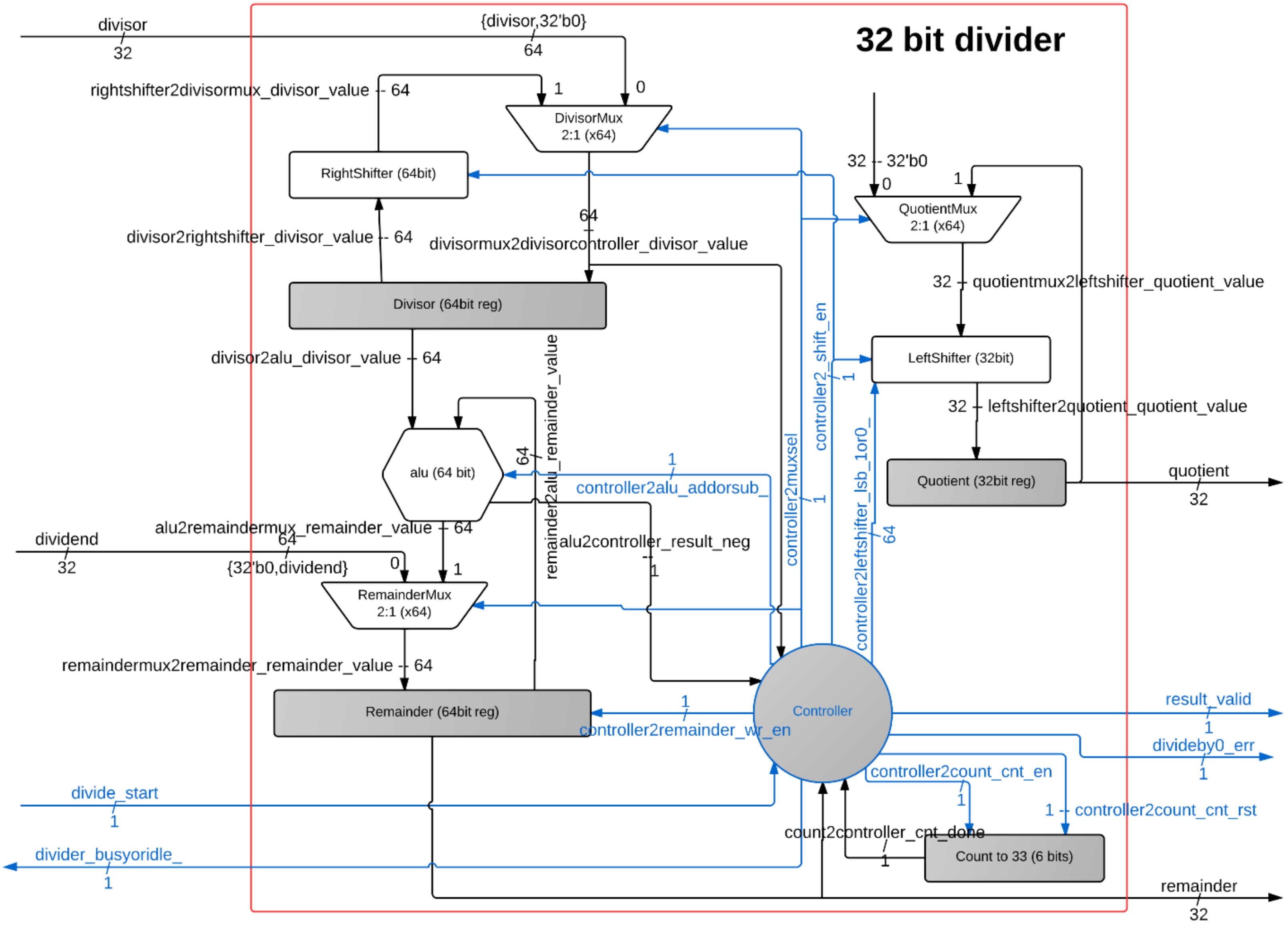
Legends

1) Boxes with white shading indicates combinational logic

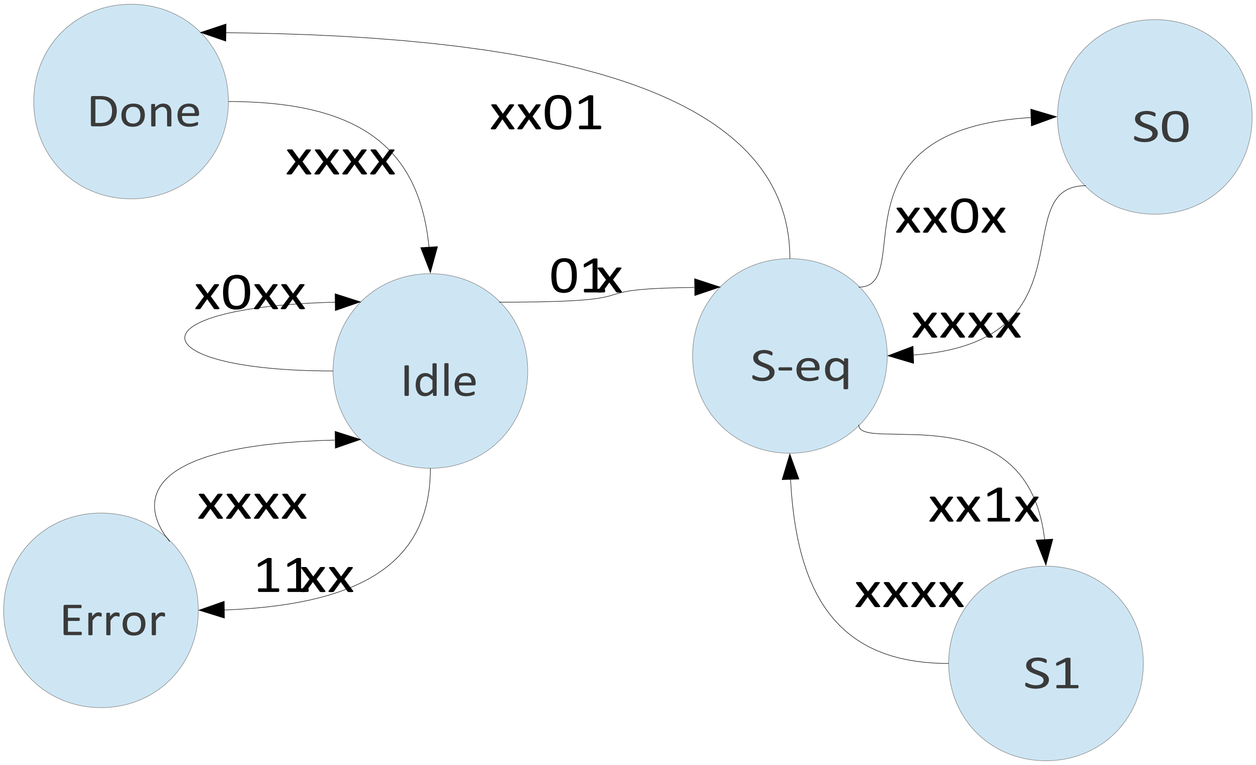
2) Boxes with gray shading indicates clocked elements (All clocked elements are fed by clk and rst which is not indicated in the schematic)

3) Black lines indicate signals in the data path

4) Blue lines indicate signals in the control path



State Machine:



Inputs: (divisor==0, divide_start, remainder<0, count2controller_cnt_done)

Problem 10 Controller Truth Table

Moore state machine's current state	State machine outputs									
	divider_busyoridle_	divideby0_err	result_valid	controller2count_cnt_en	controller2count_cnt_rst	controller2leftshifter_lsb_1or0	controller2alu_addorsub_	controller2remainder_wr_en	controller2muxsel	controller2_shift_en
Idle	0	0	0	0	1	0	x	1	0	0
Error	1	1	1	x	x	x	x	x	x	x
Done	1	0	1	x	x	x	x	x	x	x
S-eq	1	0	0	0	0	x	0	1	1	0
S0	1	0	0	1	0	1	x	0	1	1
S1	1	0	0	1	0	0	1	1	1	1