LE9.1

Bookmark this page

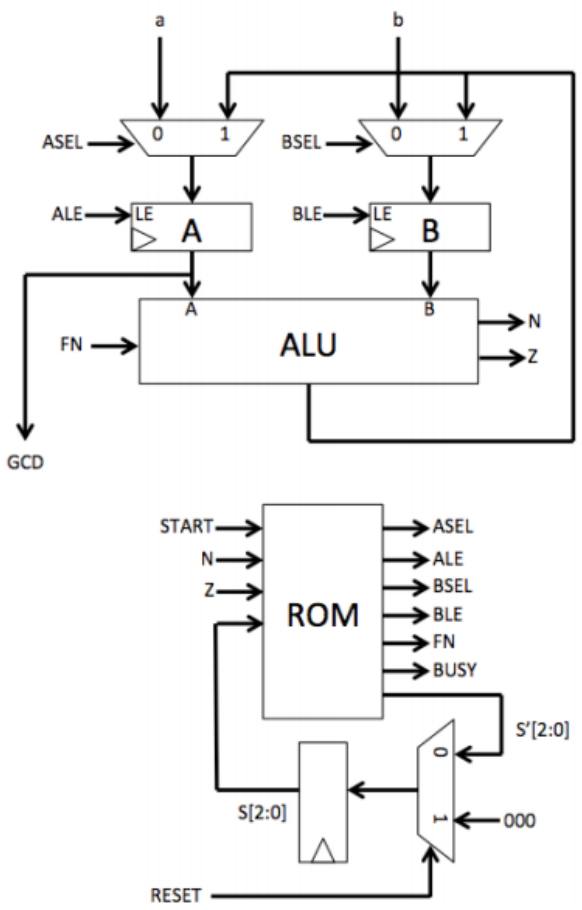
LE9.1.1: GCD

1.0/1.0 point (ungraded)

Note: you may find it helpful to view the first Worked Example in this chapter before working on this problem.

Euclid Electronic Computing Systems (EECS) has hired you as a summer intern to work on their latest product: a datapath and FSM controller that computes the greatest common divisor of two *positive* arguments represented in 32-bit two's complement form. EECS is using a variant of Euclid's algorithm, which can be written as

$$\gcd(a,b) = \begin{cases} a & a = b \\ \gcd(a-b,b) & a > b \\ \gcd(a,b-a) & a < b \end{cases}$$



The EECS circuits team has devised the datapath shown to the right, which computes GCD iteratively, holding the current values of the arguments "a" and "b" in the A and B registers respectively. The computation is initiated by placing two values of the "a" and "b" inputs to the datapath, then setting START to 1 until the BUSY output becomes 1. When BUSY becomes 0 again, the answer is available on the GCD output.

Here are the control signals for the datapath:

ASEL 0 selects the "a" input, 1 selects the output of the ALU

ALE load enable signal for the A register (1 = load)

BSEL 0 selects the "b" input, 1 selects the output of the ALU

BLE load enable signal for the B register (1 = load)

FN 0: ALU computes A-B, 1: ALU computes B-A

And here are the status signals from the datapath ALU. When the ALU function is "A-B" the N and Z status signals can be used to determine if A equals B (Z=1), A is less than B (Z=0, N=1), or A is greater than B (Z=0, N=0).

N 1 if ALU result is negative, 0 otherwise

Z 1 if ALU result is zero, 0 otherwise

An intern from the previous summer left the following unfinished table describing the contents of the ROM. Please complete the blank entries in the last two rows by entering either 0 or 1 so that the datapath correctly computes GCD according to the formula given above. If a value doesn't matter and could be either 0 or 1, enter "X", to indicate a don't-care condition.

S[2:0]	Start	N	Z	S'[2:0]	ASEL	ALE	BSEL	BLE	FN
000	0	-	-	000	0	0	0	0	0
000	1	-	-	001	0	0	0	0	0
001	-	-	-	010	0	1	0	1	0
010	-	0	0	011	1	0	1	0	0
010	-	1	0	100	1	0	1	0	0
010	-	-	1	000	1	0	1	0	0
011	-	-	-	010	1 ✔	1 ✔	X ✔	0 ✔	0 ✔
100	-	-	-	010	X ✔	0 ✔	1 ✔	1 ✔	1 ✔

Submit

Calculator

Discussion

Topic: 9. Designing an Instruction Set / LE9.1

Hide Discussion

Add a Post

Show all posts by recent activity

- Sequence of procedures — In state 100, the ALU result is negative (a < b), but the control FSM output ($BSEL = 1, BLE = 1$) is trying to load this **negative... — 17
- Can we eliminate state '010'? — I see that in state 010, the registers are loaded as per the output of state 001 initially and then as per the output of one of states 011 ... — 6
- control signals for the datapath when in state 100 — Hi, If we fall in state 100 then A-B is negative so loading the B register with the output of the ALU will end in state 010 by computing... — 7
- Load enable input for the register !! — What I know about registers that they have 3 inputs. clk, D and Q, what is load enable exactly. I don't think I have seen the circuit of t... — 2
- [deleted] — [deleted] — 4
- circuit diagrams — Hi Silvina, I'm new to this course and would like to know if there is a simple way to understand the logic behind drawing circuit diagra... — 4
- [STAFF] number of allowed attempts? — Hi Silvina, Can you please specify the number of allowed / remaining attempts per question. Or is the number of attempts unlimited? ... — 3

Previous Next

Connect

Blog
Contact Us
Help Center
Media Kit
Donate