McGill University
Department of Electrical and Computer Engineering

Course: ECSE-323 Digital System Design Winter 2008

Assignment #11 Solutions

TOPIC: Datapath/Controller System Design

Tuesday Tutorial Session

Design a datapath/controller system that computes the *arithmetic-geometric mean* of two inputs ***x*** and ***y***, using the following iterative process:

$$a_1 = \frac{x+y}{2}$$

$$g_1 = \sqrt{xy}.$$

$$a_{n+1} = \frac{a_n + g_n}{2}$$

$$g_{n+1} = \sqrt{a_n g_n}.$$

As this process is iterated many times the two values ***a*** and ***g*** will converge to the same number, which is known as the arithmetic-geometric mean.

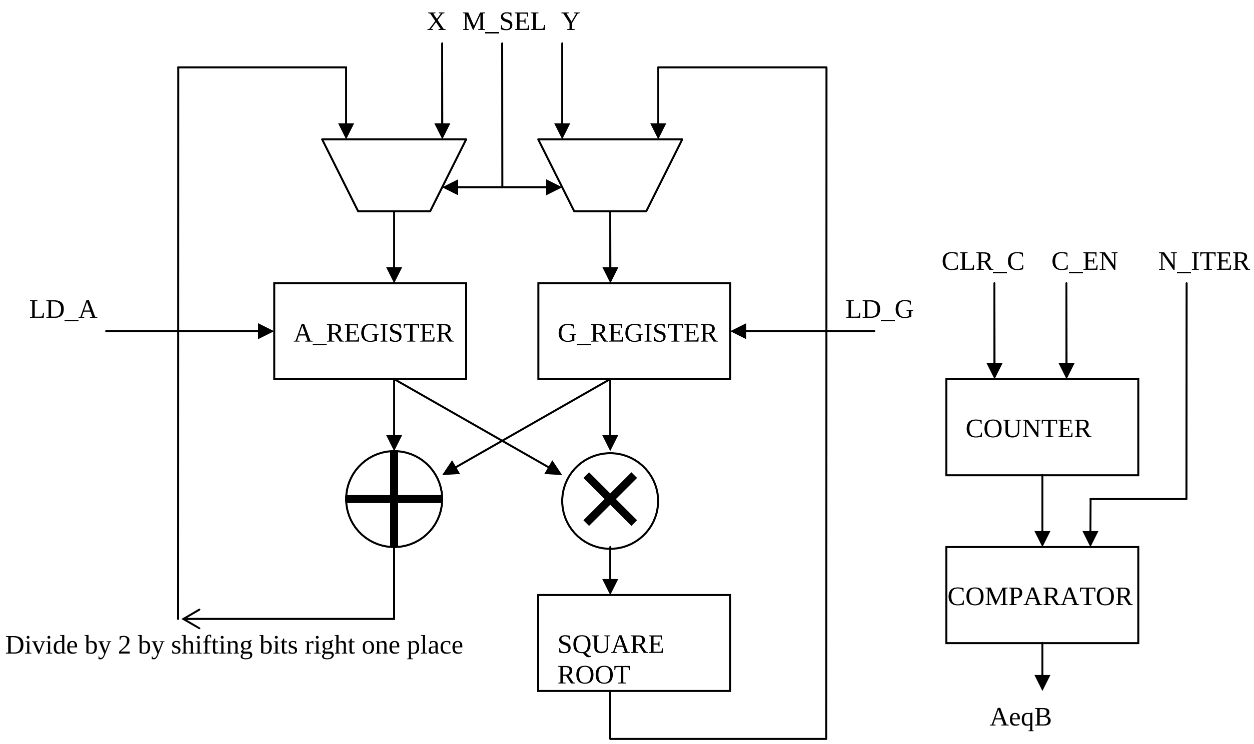
Assume that in addition to the two inputs ***x*** and ***y***, your system has an asynchronous ***START*** input, an asynchronous ***RESET*** input, and an input ***N_ITER*** indicating the number of iterations to be done. The outputs of the system should be the values ***a*** and ***g***, and a signal ***DONE*** which goes high once ***N_ITER*** iterations have been done.

a) Write down a pseudo-code description of the process to be implemented.
b) Draw the datapath, assuming that only one adder module, one multiplier module, and one square root module are available. You can use as many other modules as you see fit.
c) Draw the state transition diagram for the controller (use a Moore machine approach).

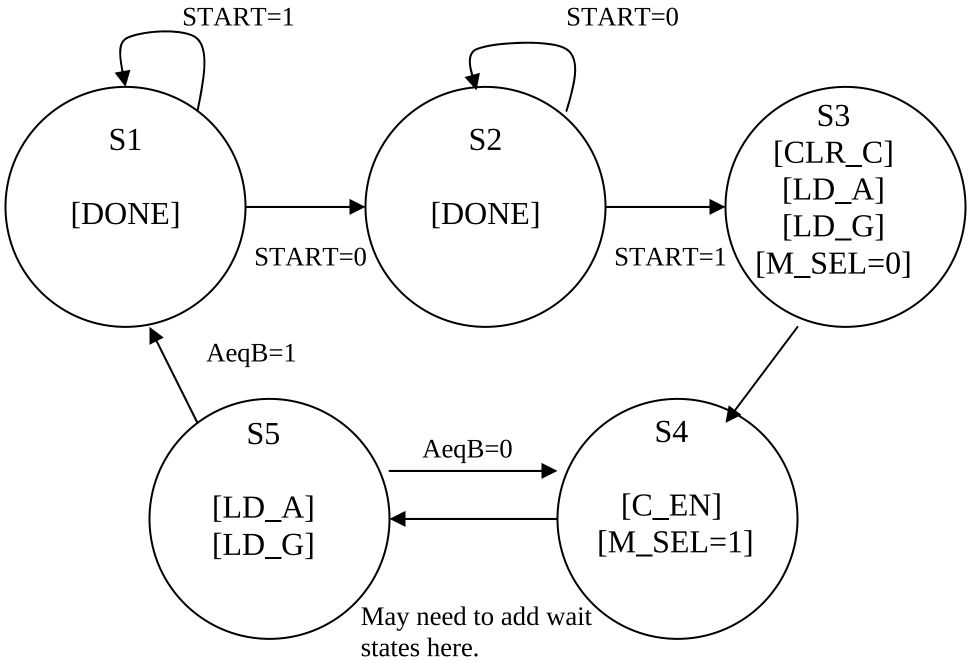
a)

1. `wait for START to go low`
2. `wait for START to go high`
3. `a = x; g = y; n=0; -- initialize`
4. `a = (a+g)/2; g = square_root(a*g); n = n+1;`
5. `if n< N_ITER go to step 4`
6. `else assert DONE and go to step 1`

b) Datapath:


X
M_SEL
Y
CLR_C
C_EN
N_ITER
LD_A
A_REGISTER
G_REGISTER
LD_G
COUNTER
COMPARATOR
Divide by 2 by shifting bits right one place
SQUARE ROOT
AeqB

c) State Diagram:



Wednesday Tutorial Session

Design a datapath/controller system that computes the natural logarithm of a number (assumed to lie in the range 1 to 2) using the following third order power series approximation:

$$\ln(1+x) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} x^n = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots \quad \text{for} \quad |x| \leq 1 \quad \text{unless} \quad x = -1$$

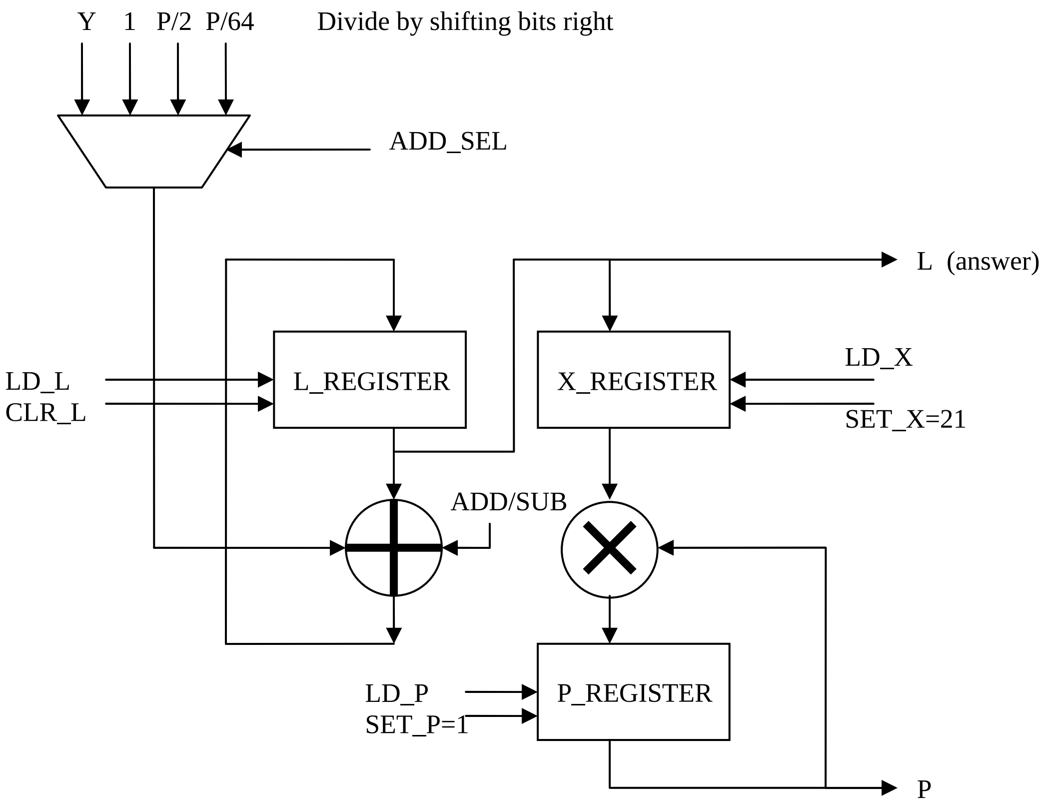
a) Write down a pseudo-code description of the process to be implemented. Assume that you only need to implement the terms shown above.
b) Draw the datapath, assuming that only one adder/subtractor module and one multiplier module are available. You can use as many other modules as you see fit. Implement the division by 3 by a multiplication by 21 and a division by 64.
c) Draw the state transition diagram for the controller (use a Moore machine approach). Your system should have as input the number ***y***, an asynchronous ***START*** signal, and an asynchronous ***reset*** signal. The system output should be the natural log of ***y*** and a ***DONE*** signal, which should go high once a valid result is available.

a)

```
1. wait for START to go low
2. wait for START to go high
3. P = 1; L = 0;
4. L = L+Y;
5. L = L-1; X = L; -- X and L are now Y-1
6. P = X*P; -- P is now X
7. P = X*P; -- P is now X^2
8. L = L-P/2; -- L is now X-X^2/2
9. P = X*P; -- P is now X^3
10. X = 21;
11. P = X*P;
12. L = L+P/64; -- L is now X-X^2/2+X^3/3
13. assert DONE and go to step 1
```

Note that the multiplier operands are always X and P. The output of the multiplier always gets loaded into P. One operand to the adder/subtractor is always L. The output of the adder/subtractor always gets loaded into L. Whether the add/sub module adds or subtracts is determined by the control signal ADD/SUB.

b) Datapath:



Y
1
P/2
P/64
Divide by shifting bits right
ADD_SEL
L (answer)
LD_L
CLR_L
L_REGISTER
X_REGISTER
LD_X
SET_X=21
ADD/SUB
LD_P
SET_P=1
P_REGISTER
P

c) State Diagram: