ECE 545 Fall 2013
Final Exam

Problem 1

Develop an ASM chart for the circuit EXAM from your Midterm Exam, described below using its

A. **pseudocode**
B. **table of input/output ports**
C. **block diagram**
D. **interface with the division into the datapath and controller.**

Please use ACTIONS (such as a0:=iv, j++, etc.) inside of your states and conditional output boxes, and CONDITIONS (such as j=8) inside of your decision boxes.

Then, redraw your ASM chart, and replace ACTIONS and CONDITIONS by names or expressions involving the corresponding outputs and inputs of the controller.

A. Pseudocode:

```
begin:
wait for s=1

a0 := iv
a1 := iv
a2 := iv
a3 := iv
k := ivp
j := 0
last_block_stored :=0

do
    if (last_block_stored /= 1) then
      wait until src_ready
      last_block_stored := last_block
      a3 := a3 ⊕ m
    end if

    for i from 0 to 63 do
        b(0) := S[a0/16] ‖ S[a0 mod 16]
        b(1) := a1 <<< S[a0 mod 16]
        b(2) := a2 * (4*a2+3) / 2^10
        b(3) := (8*a3 ⊕ 6) >> 3
        a0 := b(k mod 4)
        d0 := b(0) ⊕ b(1)
        d1 := b(2) ⊕ b(3)
        a1 := d0 · d1
        if (d0 is odd)
           a2 := d0
        else
           a2 := d1
        if ((b(3) mod 16) is prime)
           a3 := b(3)
        elsif ((b(2) mod 16) is prime)
```

```
            a3 := b(2)
         elsif ((b(1) mod 16) is prime)
            a3 := b(1)
         elsif ((b(0) mod 16) is prime)
            a3 := b(0)
         else
            a3 := ivp;
         if (k0=0)
           k := k>>1
         else
           k :=(k>>1) ⊕ (k0 << 7)
         end if
    end for

    if (last_block_stored = 1) then
       j++
    end if;

until ((last_block_stored = 1) and (j=8))

wait for s=0
y := a0 || a1 || a2 || a3
go to begin
```

Notation:

m: 8-bit message block (input)
y: 32-bit circuit output (output)
iv, ivp : 8-bit initialization vectors (constants), iv='76', ivp='0B' (in hexadecimal notation)
a0..a3, b(0)..b(3), k, d0, d1 : 8-bit intermediate values, treated as 8-bit unsigned integers
k_0: bit 0 of k

Operations:

X ⊕Y : bitwise XOR
X · Y : bitwise AND
X + Y : addition
X * Y : multiplication
X || Y: X concatenated with Y
X/16: integer part of the result of division of X by 16
X <<< Y : rotation of X to the left by the number of positions given in Y
X >>> Y : rotation of X to the right by the number of positions given in Y
X << Y : logical shift of X to the left by the number of positions given in Y
X >> Y : logical shift of X to the right by the number of positions given in Y

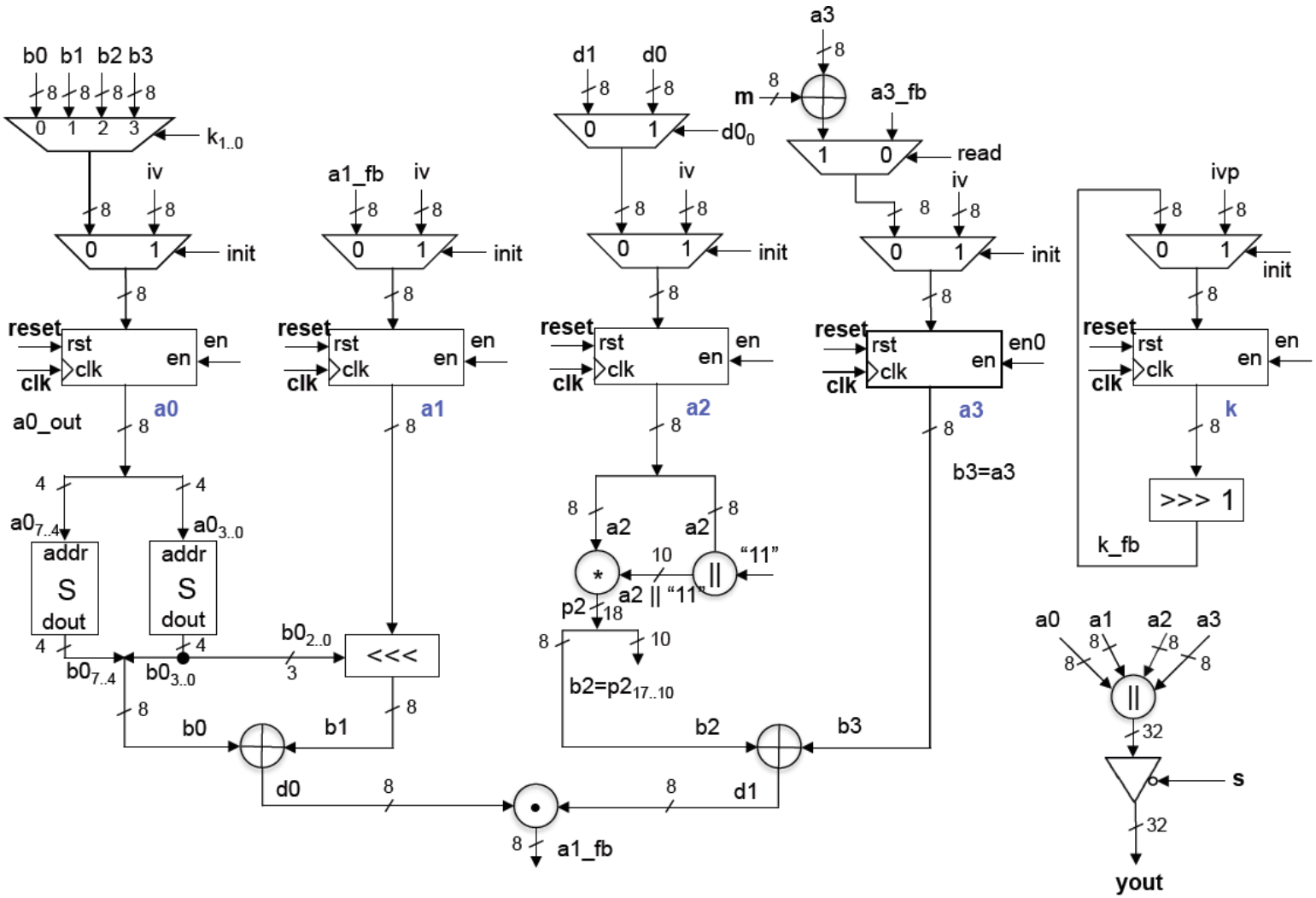
Y = S[X] : substitution defined using the following table (all values in the hexadecimal notation):

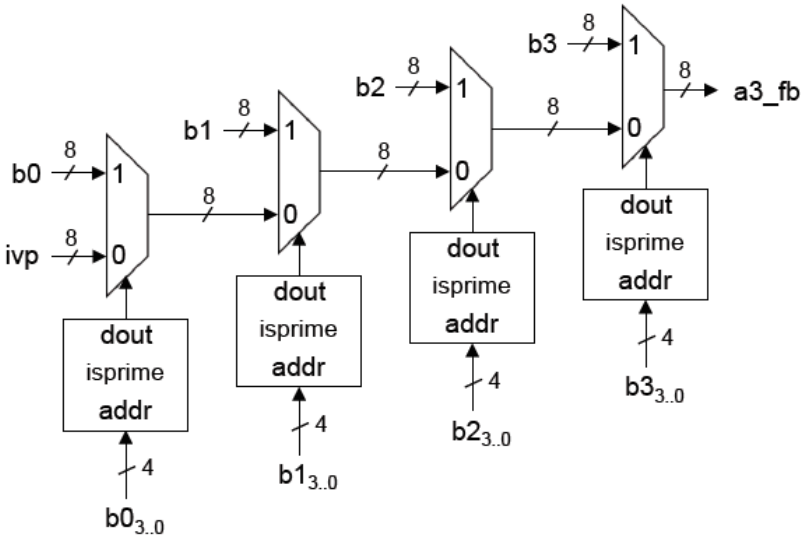
X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Y	E	D	B	0	2	1	4	F	7	A	8	5	9	C	3	6

B. Table of input/output ports:

Port	Mode	Width	Function
clk	Input	1	System clock.
reset	Input	1	Asynchronous system reset.
s		1	Operating mode: 0 = waiting for data / reading results, 1 = processing.
m	Input	8	8-bit message block.
src_ready	Input	1	Control signal indicating that the source is ready. Must remain active until source is read.
src_read	Output	1	Control signal confirming that the source was read. Active for one clock cycle.
last_block	Input	1	Control signal indicating the last block of the message.
done	Output	1	Control signal indicating that the output is ready.
yout	Output	32	32-bit output block y when s=0, high impedance otherwise.

C. Block Diagram

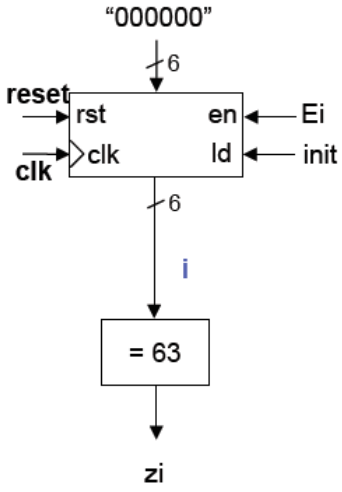


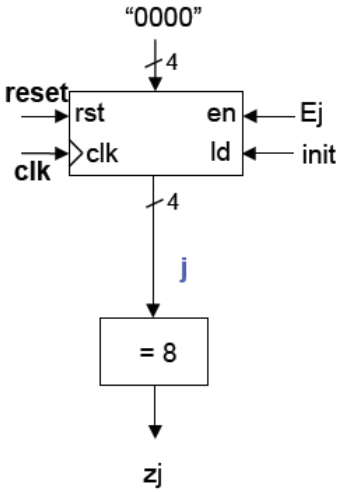


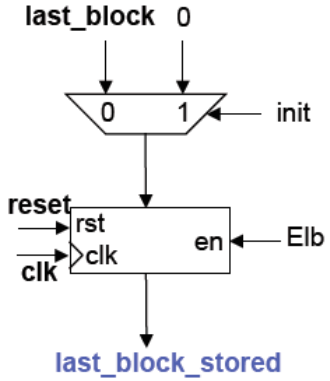
Optimizations:

b3 = (8*a3 ⊕ 6) >> 3 =
= (a3 || “110”) >> 3 = a3

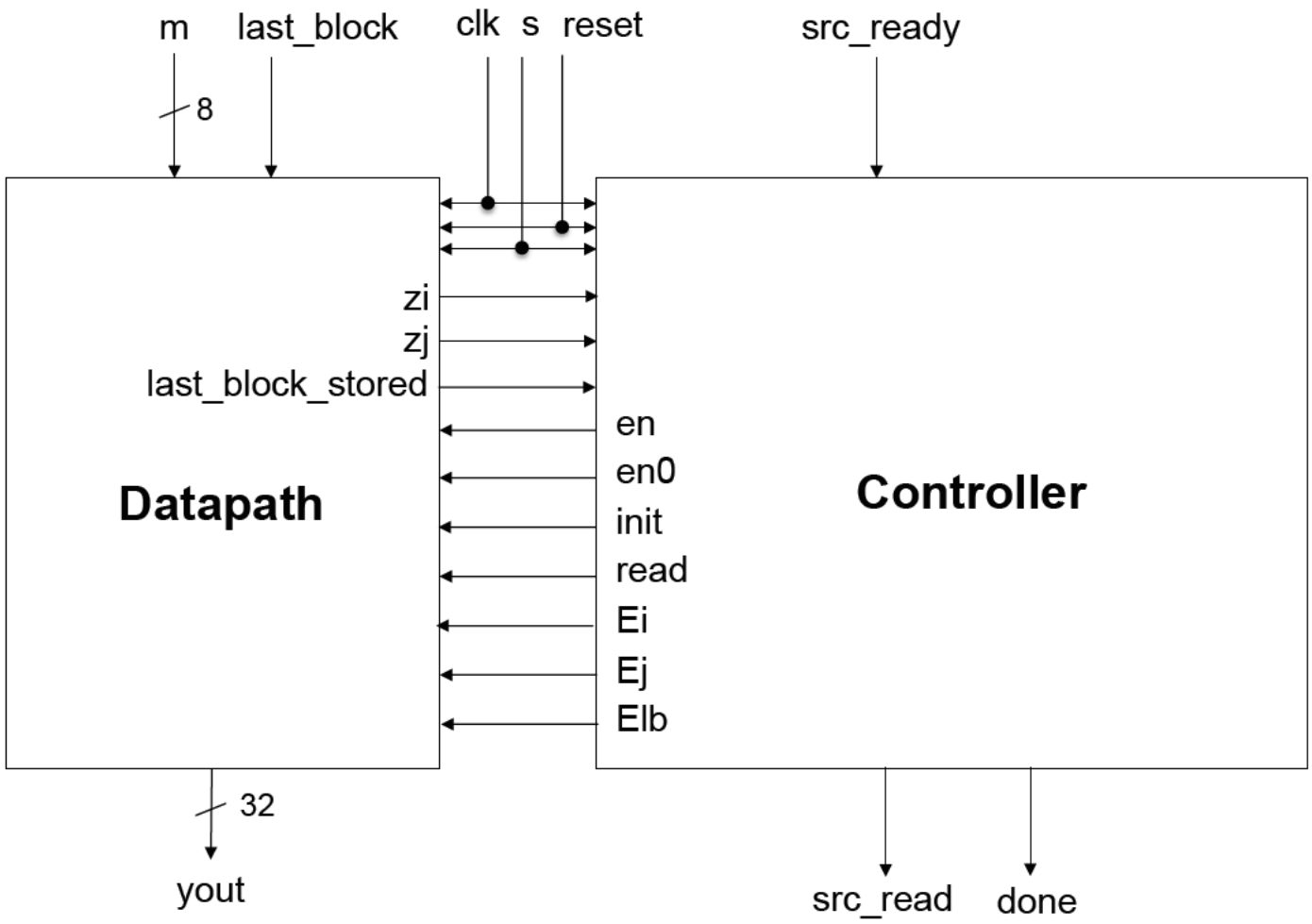
$k = 0 \,||\, k_{7..1}$ if $k_0 = 0$
$k = 1 \,||\, k_{7..1}$ if $k_0 = 1$
thus,
$k = k_0 \,||\, k_{7..1} = k >>> 1$







D. Interface with the division into the datapath and controller



Problem 2

Perform timing analysis of the circuit from Problem 1, by answering the following questions:

A. Derive formulas for

a. **Execution time as a function of the number of message blocks N, expressed in clock cycles.**

b. **Minimum Latency = Execution time for a single message block, expressed in clock cycles.**

c. **Minimum time between two consecutive input blocks (in clock cycles).**

d. **Throughput for short messages as a function of the clock period, T, and the number of message blocks N (calculated as a ratio of the total message size and the total execution time).**

e. **Throughput for long messages as a function of the clock period, T (calculated as a ratio of the message block size and the time between inputting two consecutive message blocks).**

B. Calculate values of all five parameters for T=25 ns and N=100.

C. Identify the most likely critical path in your circuit, and mark it in your block diagram (in the answer sheet).

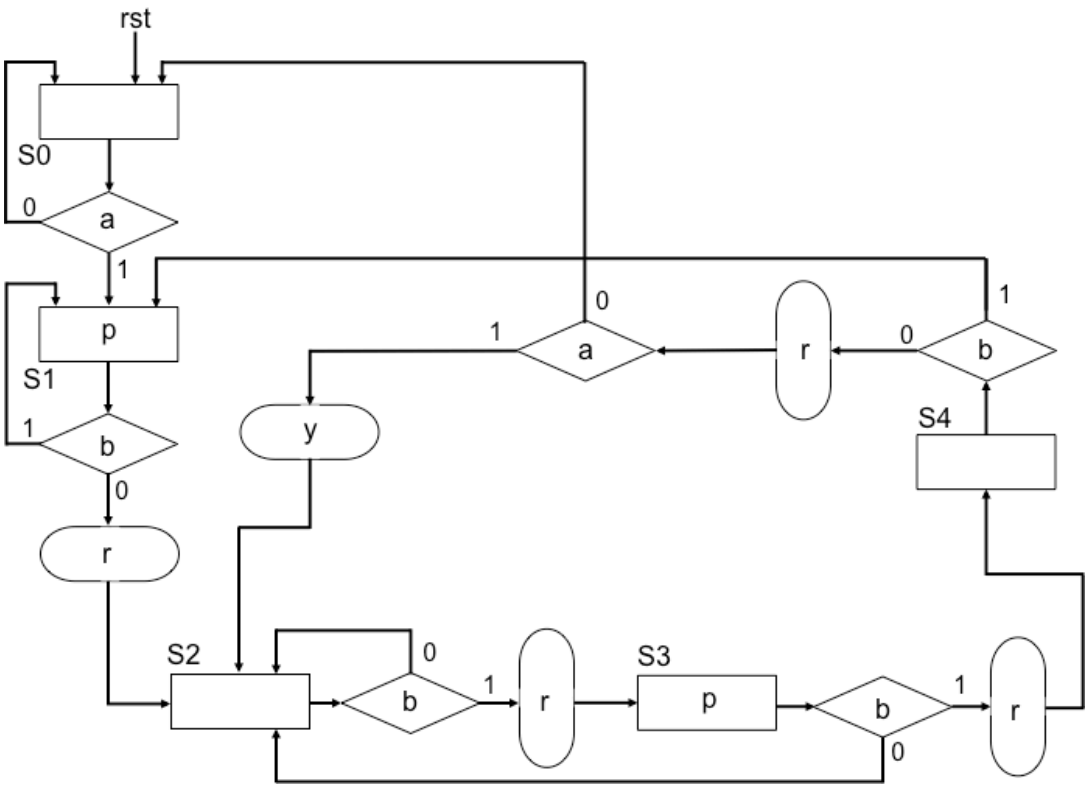
D. By how much and in what direction would the latency and throughput of this circuit changed if the clock skew between the destination register and the source register in the critical path was 2.5 ns (i.e., the clock signal arrived first at the destination register, and only 2.5 ns later at the source register)?

Justify your answer by
- drawing timing waveforms necessary to determine a new value of the clock period T', and
- providing formula for T'.

Problem 3

Assuming the controller, described using the given below ASM chart:

- supplement timing waveforms provided in the answer sheet with the values of the **state S**, and the values of the **outputs p**, **r**, and **y**
- write the VHDL dataflow code for the output function (**only**), calculating **p**, **r**, and **y**.



ASM chart of the controller.

Problem 4

Design a circuit calculating a product of two matrices, based on the following description of this operation from Wikipedia.

For two matrices:

$$\mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nm} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1p} \\ B_{21} & B_{22} & \cdots & B_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mp} \end{pmatrix}$$

where necessarily the number of columns in **A** equals the number of rows in **B**,
in this case m,
the matrix product, denoted by AB, is the $n \times p$ matrix:

$$\mathbf{AB} = \begin{pmatrix} (AB)_{11} & (AB)_{12} & \cdots & (AB)_{1p} \\ (AB)_{21} & (AB)_{22} & \cdots & (AB)_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ (AB)_{n1} & (AB)_{n2} & \cdots & (AB)_{np} \end{pmatrix}$$

where AB has entries defined as follows:

$$(AB)_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj}.$$

Make the following assumptions:

Each component of the matrices **A** and **B**, e.g., A_{11}, B_{22}, is an 8-bit unsigned integer.

$m = 2^{lm}$, e.g., $m=2^3$

$n = 2^{ln}$, e.g., $n=2^4$

$p = 2^{lp}$, e.g., $p=2^5$

Components of **A** are initially stored in **<u>RAM</u> (single port or dual port, per your preference)**, row by row, i.e., in the order

$A_{11}, A_{12}, \ldots, A_{1m}, A_{21}, A_{22}, \ldots, A_{2m}, \ldots, A_{n1}, A_{n2}, \ldots, A_{nm}$.

Components of **B** are initially stored in **<u>ROM</u> (single port or dual port, per your preference)**, row by row, i.e., in the order

$B_{11}, B_{12}, \ldots, B_{1p}, B_{21}, B_{22}, \ldots, B_{2p}, \ldots, B_{m1}, B_{m2}, \ldots, B_{mp}$.

Components of **AB** should be stored in **RAM (single port or dual port, per your preference)**, row by row, i.e., in the order

$AB_{11}, AB_{12}, \ldots, AB_{1p}, AB_{21}, AB_{22}, \ldots, AB_{2p}, \ldots, AB_{n1}, AB_{n2}, \ldots, AB_{np}$.

RAM storing Matrix **A** is initialized at the start of computations.

The circuit should be optimized for the minimum execution time.

In order to design this circuit, perform the following tasks:

Task 1

Draw a block diagram of the datapath.

Clearly specify

- **names, widths and directions of all buses**
- **names, widths and directions of all inputs and outputs of the logic components.**

Task 2

Draw an interface with the division into the Datapath and Controller.

Task 3

Draw an ASM chart using either ACTIONS and CONDITIONS or expressions involving the corresponding outputs and inputs of the controller.