ECE 545 Fall 2011
Midterm Exam

October 27, 2011

Specification:

The **EXAM** function is specified below using its:

1. Pseudocode
2. Table of input/output ports
3. Timing requirements.

1. Pseudocode:

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

do
    if (last_block_stored /= 1) then
      wait until src_ready
      last_block_stored := last_block
      a0 := a0 ⊕ m
    end if

    for i from 0 to 31 do
        b0 := S[a0 mod 16] ‖ S[a0/16]
        b1 := S[a1 mod 16] ‖ S[a1/16]
        b2 := S[a2 mod 16] ‖ S[a2/16]
        b3 := S[a3 mod 16] ‖ S[a3/16]
        a0 := ((b3 <<< 5) + b3) mod 2^8
        a1 := (b0 >> 3) ⊕ b0
        a2 := (b2 <<< (b1 mod 4)) ⊕ k
        a3 := (b3 >> (b2 mod 4)) ⊕ b3
        if (k7=0)
          k := k<<1
        else
          k :=(k<<1) ⊕ k0
        end if
    end for

    if (last_block_stored = 1) then
       y := a0
       j++
    end if;

until ((last_block_stored = 1) and (j=4))

done := 1
```

Notation:

m: 8-bit message block (input)
y: 8-bit circuit output (output)
iv : 8-bit initialization vector (constant), equal to 'C0' (in hexadecimal notation)
k0 : 8-bit constant, equal to '1B' (in hexadecimal notation)
a0..a3, b0..b3, k : 8-bit intermediate values, treated as 8-bit unsigned integers
$\mathbf{k_7}$**:** bit 7 of k

Operations:

X ⊕Y : XOR operation
X + Y : addition
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 : 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
X || Y: X concatenated with Y
X/16: integer part of the result of division of X by 16

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

2. Table of input/output ports:

Port	Mode	Width	Function
clk	Input	1	System clock.
reset	Input	1	Asynchronous system reset.
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.
y	Output	8	8-bit output block.

3. Timing Requirements:

Assume that

- **one clock cycle is used for the once-per-message initialization:**
  `a0 := iv; a1 := iv; a2 := iv; a3 := iv; k := iv`
- **one round of the main for-loop of the pseudocode executes in one clock cycle; there are a total of 32 rounds.**

Tasks:

Task 1 [40 points]: Block Diagram

Draw a block diagram of the datapath of the EXAM circuit using medium complexity components corresponding to the operations used in the pseudocode.
Your Datapath should

a) **be capable of executing the entire pseudocode given in point 1,**
b) **match interface given in point 2, and**
c) **meet all timing requirements specified in point 3.**

Clearly specify

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

Assume that one round of the main for-loop of the pseudocode executes in one clock cycle.
Minimize the number of control signals to be generated by the Control Unit.
Mark the most likely critical path in your circuit.

Task 2 [5 points]: Interface

Draw an interface of the EXAM circuit, with the division into the Datapath and the Controller. Show the names, widths, and directions of all signals forming this interface.

Task 3 [30 points]: RTL code

Write synthesizable RTL VHDL code for the portion of the Datapath responsible for execution of the main for-loop of the circuit. Include in your code any registers required for storing intermediate values. Draw a circle around the implemented portion of the block diagram.

Task 4 [10 points]: Resource Estimation

Estimate the total number of Logic Cells (1 Logic Cell = 1/2 of the CLB slice) necessary to implement the circuit from Task 3 in Xilinx Spartan 3 FPGAs.

For each line of the pseudocode inside of the main for-loop, determine the

- **number**
- **type (e.g., MLUT, Carry & Control logic (C&C), and Storage Element (SE)), and**
- **mode (e.g., logic or RAM or shift register (SR) for MLUT, flip-flop or latch for a Storage Element),**

of the corresponding FPGA resources.

Task 4 [15 points]: Testbench

Write a simple testbench for the entire EXAM circuit, using specified above interface, assuming that:

- **the message provided by the source circuit consists of 4 bytes equal to all ones**
- **the source circuit is always ready until all message words are read by the EXAM core**
- **when the last message word is read, the source circuit sets the input `last_block` to '1'**
- **the destination circuit is always ready to receive data.**