ECE 545
Fall 2014
Midterm Exam

Problem 1 [10 points]

Draw a block diagram of a simple microprocessor system, composed of

A. Microprocessor, with the bidirectional input/output DATA (8-bits), and outputs ADDR (16 bits), WR (1 bit), RD (1 bit)
B. 16k x 8 RAM0 visible by the microprocessor in the address range 0000-3FFF
C. 16k x 8 RAM1 visible by the microprocessor in the address range 4000-7FFF
D. 16k x 8 RAM2 visible by the microprocessor in the address range 8000-BFFF
E. 16k x 8 RAM3 visible by the microprocessor in the address range C000-FFFF.

Assume that a memory write cycle is indicated by the microprocessor with an active value of the output WR, and a memory read cycle with an active value of the output RD.

Problem 2 [15 points]

Fill in the blanks in the code of the Debouncer circuit, **provided in the answer sheet**. **Do not write this code from scratch**! The block diagram of the Debouncer is shown in Fig. 1.

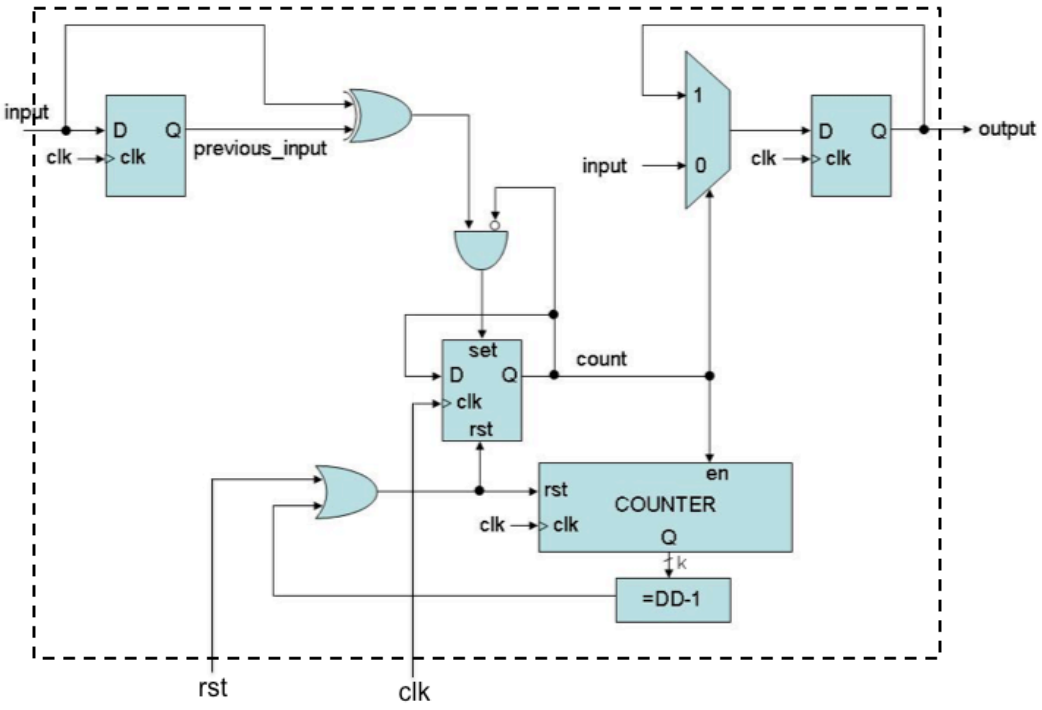


Fig. 1: Block diagram of the Debouncer circuit.

Problem 3 [15 points]

Write a complete simple testbench capable of testing the debouncer shown in Fig. 1, by applying

A. clk, B. rst, C. input shown in Fig. 2.

- The clock signal, clk, should be a periodical signal, with the period of 10 ns.
- The reset signal, rst, should be a non-periodical signal, active high for the first 50 ns of the simulation period.
- The input signal should look as shown in Fig. 2, and should have all changes happening on the falling edges of the clock.
- The Debouncer, shown in Fig. 1, should be instantiated with the following values of the generics k and DD: k = 4, DD = 15.

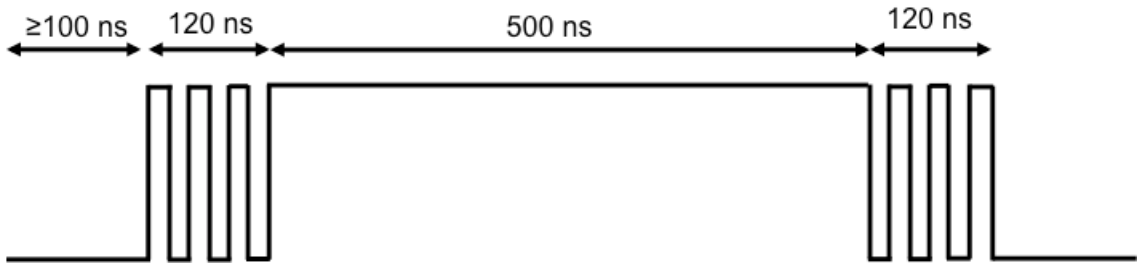


Fig. 2: Timing waveform of the input signal.

Problem 4 [60 points]

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

a. Pseudocode
b. Table of input/output ports
c. Functional and timing requirements.

1. Pseudocode:

DES Encryption, $C=E_K(M)$:

```
X = IP(M)
L(0) = X1..32
R(0) = X33..64
for i = 0 to 15 do
      L(i+1) = R(i)
      R(i+1)= L(i)⊕ f(R(i), K(i))
end for
Y = R(16) || L(16)
C = IP-1(Y)
```

DES Decryption, $M=D_K(C)$:

```
X = IP(C)
R(16) = X1..32
L(16) = X33..64
```

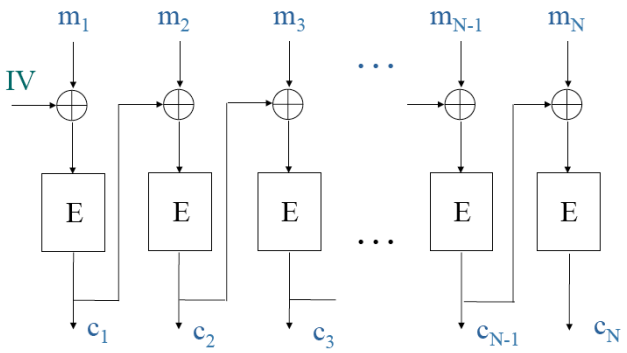
```
for i = 15 downto 0 do
      R(i) = L(i+1)
      L(i)= R(i+1)⊕ f(L(i+1), K(i))
end for
Y = L(0) || R(0)
M = IP^-1(Y)
```

Function f, Z = f(R, K):

```
U = E(R) ⊕ K
for i = 0 to 7 do
      V(i) = U_i*6+1..i*6+6
      W(i) = Sbox(V(i))
end for
Z = W(0) || W(1) || W(2) || W(3) || W(4) || W(5) || W(6) || W(7)
```

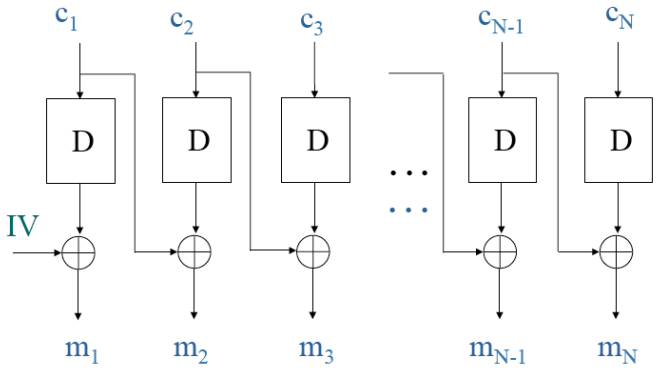
Both encryption and decryption should be able to operate in the CBC mode, shown conceptually in the diagrams below:

Cipher Block Chaining Mode - CBC Encryption



$c_i = E_K(m_i \oplus c_{i-1})$ for i=1..N c_0=IV

Cipher Block Chaining Mode - CBC Decryption



$m_i = D_K(c_i) \oplus c_{i-1}$ for i=1..N c_0=IV

Notation:

M: 64-bit message block, with bits arranged as follows $M_1M_2\ldots M_{64}$
C: 64-bit ciphertext block, with bits arranged as follows $C_1C_2\ldots C_{64}$
X, Y: 64-bit intermediate variables
L(i), R(i): 32-bit intermediate variables
U: 48-bit intermediate variable
V(i): 6-bit intermediate variables
W(i): 4-bit intermediate variables
K(i): 48-bit round key K(i)

Operations:

X ⊕Y : bitwise XOR
X || Y: X concatenated with Y
IP(), IP^{-1}(): fixed permutations
Y=E(X): an expansion function replacing a 32-bit vector X with a 48-bit vector Y
Y=Sbox(X): a 6x4 Sbox, i.e., a substitution function with a 6-bit input and a 4-bit output.

2. 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 = loading round keys/ waiting for data, 1 = processing.
decrypt	Input	1	Control signal: decrypt=0 represents encryption, decrypt=1 represents decryption.
IV	Input	64	64-bit Initialization Vector.
write_IV	Input	1	Synchronous write control signal for the input IV.
Xi	Input	64	64-bit input block (message block M for encryption, ciphertext block C for decryption).
write_Xi	Input	1	Synchronous write control signal for the input Xi.
last_block	Input	1	Control signal indicating the last block of the message.
Kj	Input	48	Round key K(j).
j	Input	4	Index of the round key K(j), j=0..15.
write_Kj	Input	1	Synchronous write control signal for the round key K(j).
done	Output	1	Control signal indicating that the output block is ready.
Yi	Output	64	64-bit output block (ciphertext block C for encryption, message block M for decryption) when s=0, high impedance otherwise.

3. Functional and timing requirements:

Assume that

- **When s=0, round keys are loaded to the internal RAM. When s=1, the circuit is ready to perform encryption or decryption, depending on the values of control signals.**
- **Loading round keys to internal memory takes 16 clock cycles.**

- **Encryption of one message block takes 16 clock cycles.**
- **Decryption of one ciphertext block takes 16 clock cycles.**

Tasks:

Task 1: Block Diagram [45 points]

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 functional and 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.

Minimize the number of control signals to be generated by the Control Unit.

Task 2: Substitution Sbox [5 points]

Explain in detail your implementation of the substitution function Y=Sbox(X). Which logic component do you use to implement this function? How is this component initialized?

Task 3: Interface [10 points]

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

Problem 5 [bonus 10 points]

Write VHDL code describing the following functionality:

$C = (A << 1) \oplus (A_{127} \cdot \text{'8C9'})$

where

A and **C** are 128-bit buses,
$\mathbf{A_{127}}$ is the most significant bit of A,
'8C9' is a constant expressed in the hexadecimal representation,
$\mathbf{X} \oplus \mathbf{Y}$ is a bitwise XOR of vectors X and Y,
$\mathbf{x} \cdot \mathbf{C}$ is a multiplication of the binary variable x by the constant C (equal to C if x=1, and equal to 0 otherwise).