Middle East Technical University

# CENG 232

Logic Design

Spring 2017-2018

Lab-4 Part-1

**Part 1 Due date:** Wednesday, April 18, 2018, 23:59
Late submissions are not allowed

## 1 Function Operations (Individual Work)

**This part of the lab will be performed and submitted individually.**

In this part, you are expected to implement basic memories as 2 Verilog modules. These modules will be used to store the functions, evaluate and store the result of evaluations.

### 1.1 Problem Definition

Each function will be in the form: $a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0x^0$. The module FUNCROM will contain 16 registers. Each register holds a 5-bit binary number which represents the coefficients of the function starting from $a_4$ through $a_0$. The values of $a_4$ through $a_0$ can either get +1 or -1: These values are represented in a register as 0 and 1, respectively. Examples of 3 registers and the functions they represent are given in the below table:

| Register Value | Function |
|---|---|
| 00000 | $x^4 + x^3 + x^2 + x + 1$ |
| 01010 | $x^4 - x^3 + x^2 - x + 1$ |
| 11110 | $-x^4 - x^3 - x^2 - x + 1$ |

Given the index value as input, FUNCROM module basically returns the register value stored at that index as output. It works as a combinational circuit. The values of FUNCROM will be as the following table:

| Index | Register Value |
|---|---|
| 0000 | 00000 |
| 0001 | 00010 |
| 0010 | 00100 |
| 0011 | 00111 |
| 0100 | 01010 |
| 0101 | 01011 |
| 0110 | 01101 |
| 0111 | 01110 |
| 1000 | 10001 |
| 1001 | 10010 |
| 1010 | 10100 |
| 1011 | 10111 |
| 1100 | 11001 |
| 1101 | 11010 |
| 1110 | 11110 |
| 1111 | 11111 |

The module FUNCRAM will contain 16 registers. Each register contains 9 bits. Initially, the values of all RAM registers will be 9'b000000000. FUNCRAM module basically has two functionalities: 1) Reading data in given index, 2) Writing data into memory. These are determined by the modes given below:

- mode(1 bit):
  **0** → read mode
  **1** → write mode

**Write Mode**
In write mode, your task is to evaluate the modulo 7/derivative of the function with the given argument and store the result in the FUNCRAM. The input will be a 7-bit number. The first 4 bits of the input are used as an index to select the functions from the FUNCROM. The fifth bit is used to determine the type of the operation. If the value of fifth bit is 0, the function with the given the argument will be evaluated which is followed by mod 7 operation and the final result is calculated. If the value of fifth bit is 1, the derivative of the function will be evaluated with the given argument. The last 2 bits of the input correspond to the argument of the function where binary 00, 01, 10 and 11 will represent +2,+1,-1 and -2 in decimal, respectively. The output will be a 9-bit number where the most significant bit denotes the sign (0 if positive and 1 if negative). The examples are given in the below table:

| Input | ROM Index | Operation Type | Arg. | Function At Given ROM Index | Calculation | Output |
|---|---|---|---|---|---|---|
| 0000000 | 0000 | modulo 7 | 2 | $P(x) = x^4 + x^3 + x^2 + x^1 + 1$ | $P(2) = 2^4 + 2^3 + 2^2 + 2 + 1 = 31$<br>$P(2)$ modulo $7 = 31 \bmod 7 = 3$ | 000000011 |
| 0000101 | 0000 | derivative | 1 | $P(x) = x^4 + x^3 + x^2 + x^1 + 1$ | $P'(x) = 4x^3 + 3x^2 + 2x + 1$<br>$P'(1) = 4 + 3 + 2 + 1 = 10$ | 000001010 |
| 1111100 | 1111 | derivative | 2 | $P(x) = -x^4 - x^3 - x^2 - x^1 - 1$ | $P'(x) = -4x^3 - 3x^2 - 2x - 1$<br>$P'(2) = -32 - 12 - 4 - 1 = -49$ | 100110001 |

The output will be stored in a register of FUNCRAM. The index of that register is the first four bits of the input. Please note that multiple write operation on the same index will overwrite the previous value in that index. The operation will be done with the rising edge of the clock pulse.

**Read Mode**
In read mode, the value in given index location of FUNCRAM will be returned as output. There will be no write operation to the FUNCRAM in this mode. This operation is combinational, and is **not** triggered by a clock pulse.

The FUNCROM and FUNCRAM modules will be used by an upper module called FUNCMEMORY. FUNCMEMORY is an interface for FUNCROM and FUNCRAM modules. The module definition of FUNCMEMORY is given below:

**module FUNCMEMORY(input mode, input [6:0] memInput, input CLK, output wire [8:0] result)**

where **mode** represents FUNCRAM read/write mode, **memInput** represents the 7-bit input used in FUNCRAM and FUNCROM modules and **result** represents the 9-bit output stored in FUNCRAM.

The module definitions of FUNCROM and FUNCRAM are as follows:

**module FUNCROM (input [3:0] romAddress, output reg[4:0] romData)**

where **romAddress** represents the index of ROM register and **romData** represents the register value given the romAddress value.

**module FUNCRAM (input mode,input [3:0] ramAddress, input [4:0] dataIn,input op, input [1:0] arg, input CLK, output reg [8:0] dataOut)**

where **mode** represents read/write mode, **ramAddress** represents the index of the FUNCRAM register to store the output, **dataIn** represents the coefficient values of the function, **op** represents the type of operation (i.e. modulo 7/derivative of the function), **arg** represents the 2-bit argument of the function and **dataOut** represents the 9-bit output stored in FUNCRAM.

You will implement only FUNCROM and FUNCRAM modules. FUNCMEMORY module will be provided for you and will not be implemented.

## Deliverables

- Implement both modules in a single Verilog file: *lab4_1.v*. Do NOT submit your test-benches. You can share your test-benches on the newsgroup.
- Submit the file through the COW system before the given deadline. **April 18, 2018, 23:59hrs.**
- This part is supposed to be done individually, not with your partner. Any kind of cheating is not allowed.