Saif Ba Madhaf - 527028

VHDL Week 5

Questions:

Q1. Is it possible to use more than one Architecture with one Entity?

Yes, it can have multiple entities, however it must be associated with an entity.

Q2. When do you use a configuration?

It is used to associate an architecture with an entity.

Q3. What are the resources of libraries?

- STANDARD package.
- IEEE developed packages.
- Altera component packages
- Any LIVRARY of design unit that is referenced in a design.

Q4. Why do you have to include the library IEEE?

Because it contains VHDL packages that you might want to use like STD_LOGIC and NUMERIC_STD.

Q5. When you have a multidrive for one signal with std_logic signals and you give them a ‘0’and a ‘1’ what will be the resolved signal in the simulation?

It will give you the value X as it is unknown. Because it cannot receive both a 0 and 1 at the same time.

Q6. What is a signal, and what is the difference with a variable in VHDL?

Signals are used as an interconnect (wire) to connect between two processes.

Variables are basically like local storage within a process.

Q7. Rewrite “temp <= x”AA” “ to an assignment with a logic std_vector.

temp <= x & "AA";

Q8. How does the “With .. select“ assignment work? Give an example.

HOW?

WITH <expersion> SELECT

<signal_name> <= <signal/value> WHEN <condition_1>,

<signal/value> WHEN <condition_2>,

...

<signal/value> WHEN OTHERS;

EXAMPLE:

With sel SELECT
q<= a WHEN “00”,

b WHEN “01”,

c WHEN “10”,

d WHEN OTHERS;

Q9. What is the difference between an “inertial delay” and a “Transport

delay”?

Inertial Delay: A pulse that is short of the propagation delay will not be transmitted.

Transport Delay: Any pulse is transmitted no matter how short the suration is.

Q10. Explain what “process (all)“ means in VHDL.

Its option to represents all the inputs in the process without the need of mentioning them individually.

Q11. Can you use a “wait“ statement in a “process?

Yes, you can use wait statements in a process, in fact you can have multiple in a process.

Q12. Can a “case” statement be used without an explicit “process” statement?

No, it cannot be used without a process in VHDL, except simple signal assignments.

Q13. Can synthesis tools support all wait statements?

No, wait statements are limited in synthesis.

Q14. Can you use Variables outside explicit processes?

No, variables are only declared inside a process.

Q15. Watch the video example at (1:10:54) . Describe in VHDL a memory for a black and white picture of 10x10 pixels.

```
ARCHITECTURE logic OF blackandwhiteMemory IS

TYPE mem IS ARRAY (0 to 9) OF std_logic_vector(9 DOWNTO 0);

SIGNAL mem_10x10_b, mem_10x10_w : mem;

BEGIN

BEGIN
  PROCESS
  BEGIN
    FOR i IN 0 TO 9 LOOP
```

```
        FOR j IN 0 TO 9 LOOP
            mem_10x10_b(i, j) <= '0';  -- makes all black
        END LOOP;
    END LOOP;
    WAIT;
  END PROCESS;

  -- Initialize memory for white pixels
  PROCESS
  BEGIN
    FOR i IN 0 TO 9 LOOP
        FOR j IN 0 TO 9 LOOP
            mem_10x10_w(i, j) <= '1';  -- makes all white
        END LOOP;
    END LOOP;
    WAIT;
  END PROCESS;
END ARCHITECTURE logic;
```

Q16. Explain “enumerated type declaration”.

It allows you to create your own data type name and values. It is used in state machines and in making code more readable.

HOW?

TYPE <your data type> IS

(Data type items or values separated by commas);

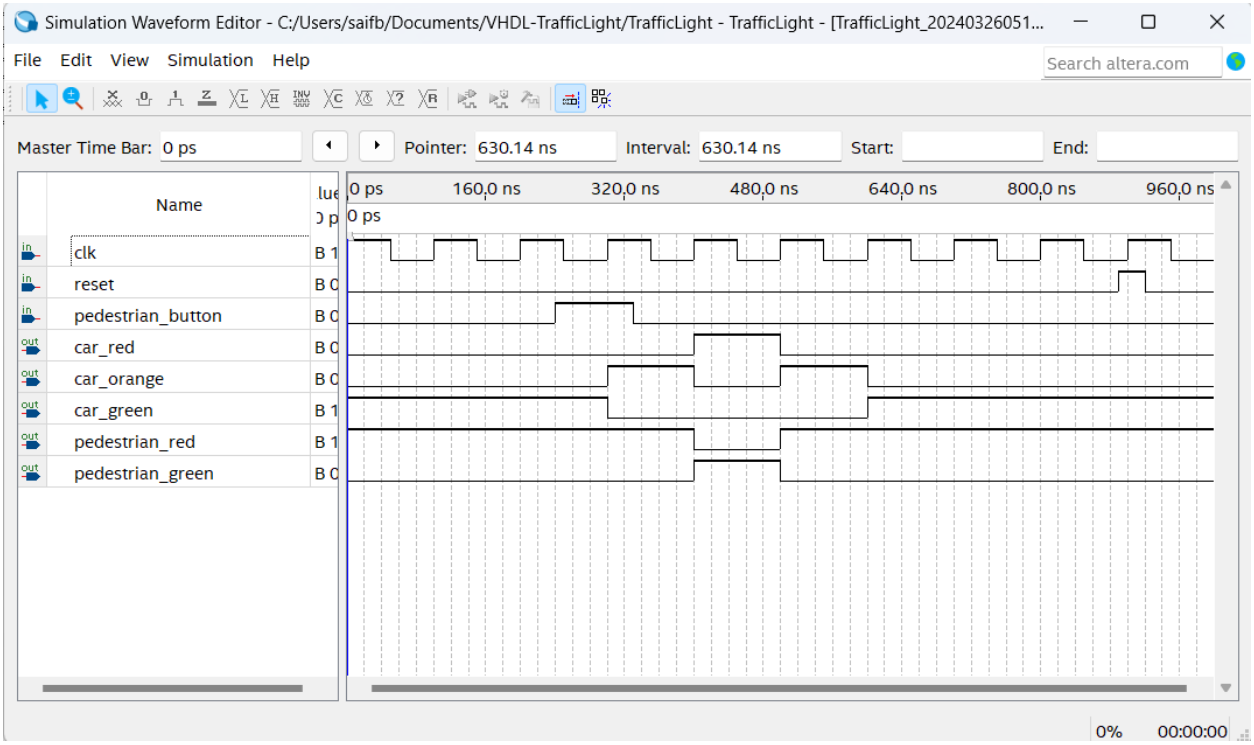
Self-test chapter 11:

1. Point in the VHDL of the state machine which processes are designed.

There are three states (A,B, and C) each have defined outputs. There is a reset input where if its 1 it returns to state A and if there is a rising edge in the clock there are three cases for each of the states with these state transitions the outcome is a state machine.

Exercise chapter 11:

1. **Write in VHDL and simulate a traffic light controller for a pedestrian crossing. Assume the car lights are initially green, with an input for the pedestrian, they can initiate a , orange - red - green for the cars and a green - red, for them self.**



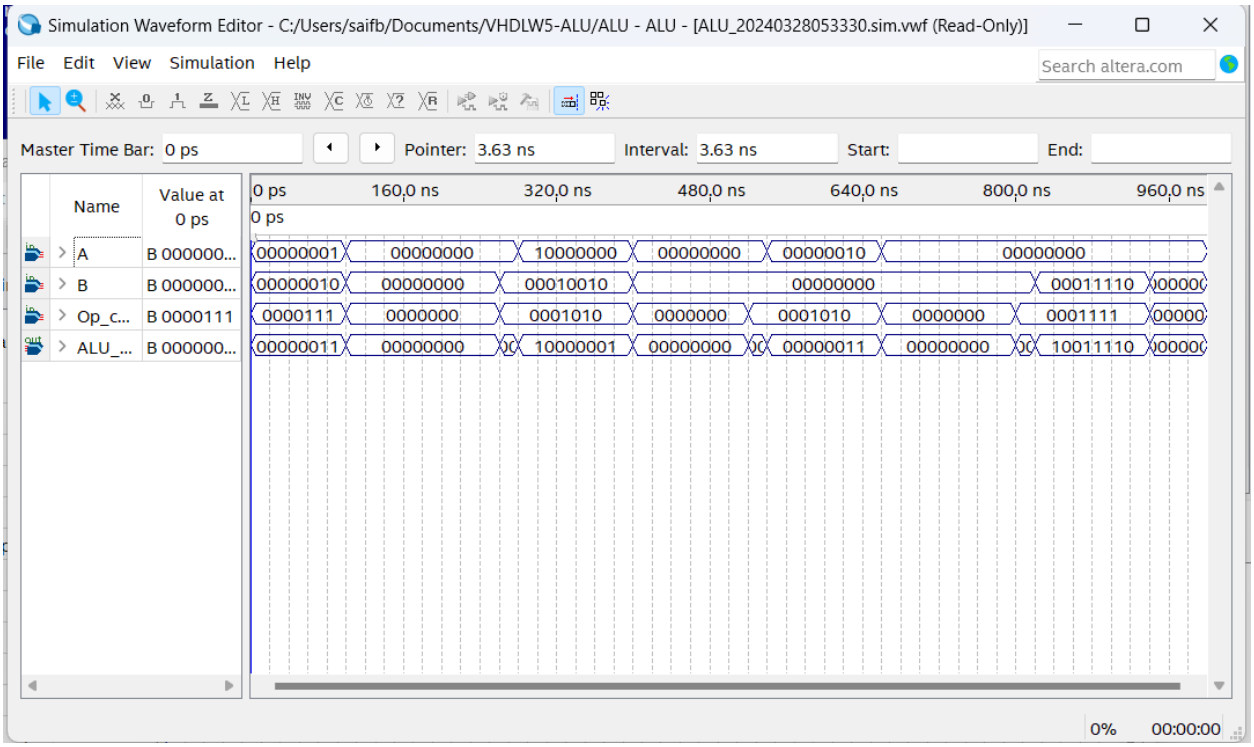
Self-test chapter 12:

1. **Is an ALU an Sequential or a logical piece of hardware?**

The ALU is a logical piece of hardware as it does not rely on previous input like sequential. For example, in logical piece of hardware what you input gets output depending on the instruction given without looking at previous input. However, in sequential it depends on previous input such as Flip Flop.

Exercise chapter 12:

1. Design an ALU with a subset instruction set as the PIC16F684. (Take only the 7 MSB from the op_code) op_code for (ADDWF,ANDWF, CLRF,INCF,MOVF, and BSF)



As we can see, it looks a bit complicated. To begin with we two inputs A, B, and opcode. A and B are the operations. Opcode is the code to know what instruction we are going to use (look below for reference). For example, in the first situation we are using ADDWF instruction (opcode - 00000111) to add both A and B, we can see the calculation is correct in output ALU. One more example would be the last one which is the BSF how it works is u set bit b in register F to 1. (opcode - 00001111) we can assume that in our case bit b is the msb in our byte. We can see that alu output for that instruction is correct. 10011110 it successfully set bit b to 1.

PIC16F684

TABLE 13-2: PIC16F684 INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb	LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d	Add W and f	1	00 0111 dfff ffff		C, DC, Z	1, 2
ANDWF	f, d	AND W with f	1	00 0101 dfff ffff		Z	1, 2
CLRF	f	Clear f	1	00 0001 lfff ffff		Z	2
CLRW	–	Clear W	1	00 0001 0xxx xxxx		Z	
COMF	f, d	Complement f	1	00 1001 dfff ffff		Z	1, 2
DECF	f, d	Decrement f	1	00 0011 dfff ffff		Z	1, 2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00 1011 dfff ffff			1, 2, 3
INCF	f, d	Increment f	1	00 1010 dfff ffff		Z	1, 2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00 1111 dfff ffff			1, 2, 3
IORWF	f, d	Inclusive OR W with f	1	00 0100 dfff ffff		Z	1, 2
MOVF	f, d	Move f	1	00 1000 dfff ffff		Z	1, 2
MOVWF	f	Move W to f	1	00 0000 lfff ffff			
NOP	–	No Operation	1	00 0000 0xx0 0000			
RLF	f, d	Rotate Left f through Carry	1	00 1101 dfff ffff		C	1, 2
RRF	f, d	Rotate Right f through Carry	1	00 1100 dfff ffff		C	1, 2
SUBWF	f, d	Subtract W from f	1	00 0010 dfff ffff		C, DC, Z	1, 2
SWAPF	f, d	Swap nibbles in f	1	00 1110 dfff ffff			1, 2
XORWF	f, d	Exclusive OR W with f	1	00 0110 dfff ffff		Z	1, 2
BIT-ORIENTED FILE REGISTER OPERATIONS							
BCF	f, b	Bit Clear f	1	01 00bb bfff ffff			1, 2
BSF	f, b	Bit Set f	1	01 01bb bfff ffff			1, 2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01 10bb bfff ffff			3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01 11bb bfff ffff			3

Exercise chapter 14:

1. Try the example and try if this size of memory fits on the used chip on the danjel board.

Flow Status	Successful - Tue Mar 26 14:54:49 2024
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	MemoryTest
Top-level Entity Name	MemoryTest
Family	MAX 10
Device	10M08SAE144C8GES
Timing Models	Preliminary
Total logic elements	1 / 8,064 (< 1 %)
Total registers	0
Total pins	24 / 101 (24 %)
Total virtual pins	0
Total memory bits	512 / 387,072 (< 1 %)
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 1 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 1 (0 %)

When I ran the code into the system, it said that it took less than 1 percent of the system's memory 512/387,072 as shown above. It basically means that there is more than enough memory.

References

https://ww1.microchip.com/downloads/en/devicedoc/41202D.pdf

https://www.youtube.com/watch?v=zm-RA6BsYmc