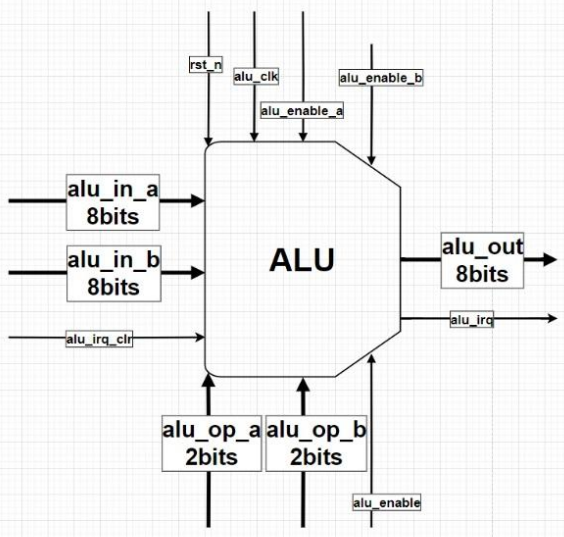
Digital IC Verification Track - New Capital Branch

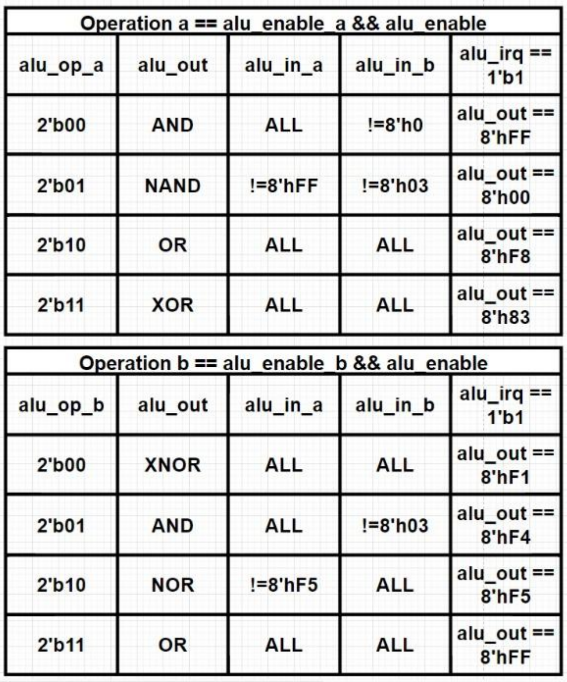
Assignment 1: Design Requirements Extraction

Submitted to: Eng. Mahmoud Saeed Elbosily

Group 2

Ahmed Ashraf Mohamed El-Hady
Tasneem Abo El-Esaad Abd El-Razik
Youssef Nasser Abd El-Aal Abd El-Fattah





Operation a == alu_enable_a && alu_enable				
alu_op_a	alu_out	alu_in_a	alu_in_b	alu_irq == 1'b1
2'b00	AND	ALL	!=8'h0	alu_out == 8'hFF
2'b01	NAND	!=8'hFF	!=8'h03	alu_out == 8'h00
2'b10	OR	ALL	ALL	alu_out == 8'hF8
2'b11	XOR	ALL	ALL	alu_out == 8'h83

Operation b == alu_enable_b && alu_enable				
alu_op_b	alu_out	alu_in_a	alu_in_b	alu_irq == 1'b1
2'b00	XNOR	ALL	ALL	alu_out == 8'hF1
2'b01	AND	ALL	!=8'h03	alu_out == 8'hF4
2'b10	NOR	!=8'hF5	ALL	alu_out == 8'hF5
2'b11	OR	ALL	ALL	alu_out == 8'hFF

Design Requirements Extraction

- **alu_in_a**
  - Synchronized to alu_clk, positive edge trigger.
  - Constrained random:
    - When ***alu_op_a*** == NAND, ***alu_enable*** == 1, ***alu_enable_a*** == 1 and ***alu_enable_b*** == 0 then ***alu_in_a*** can't take value of ***8'hff.***
    - When ***alu_op_b*** == NOR, ***alu_enable*** == 1, ***alu_enable_a*** == 0 and ***alu_enable_b*** == 1 then ***alu_in_a*** can't take a value of ***8'hf5.***
- **alu_in_b**
  - Synchronized to alu_clk, positive edge trigger.
  - Constrained random:
    - When ***alu_op_a*** == AND, ***alu_enable*** == 1, ***alu_enable_a*** == 1 and ***alu_enable_b*** == 0 then ***alu_in_b*** can't take value of ***8'h0.***
    - When ***alu_op_a*** == NAND, ***alu_enable*** == 1, ***alu_enable_a*** == 1 and ***alu_enable_b*** == 0 then ***alu_in_b*** can't take value of ***8'h03.***
    - When ***alu_op_a*** == AND, ***alu_enable*** == 1, ***alu_enable_a*** == 0 and ***alu_enable_b*** == 1 then ***alu_in_b*** can't take value of ***8'h03.***
- **rst_n**
  - Active low, asynchronous reset.
  - Whenever ***rst_n*** is low, then ***alu_out*** and ***alu_irq*** should be driven to ***0***.
  - ***rst_n*** should be asserted at least once at start-up.
- **alu_clk**
  - Positive edge triggered clock.
  - ***alu_clk*** has a frequency = 32 MHz.

- **alu_enable**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - If ***alu_enable*** is de-asserted:
    - All outputs should be maintained, if ***rst_n*** is high, regardless of ***alu_enable_a*** or ***alu_enable_b***.
  - If ***alu_enable*** is asserted:
    - ***alu_enable_a*** and ***alu_enable_b*** can't be asserted at the same time.
- **alu_enable_a and alu_enable_b**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - When ***alu_enable*** is asserted:
    - if ***alu_enable_a*** == 0 and ***alu_enable_b*** == 0, it is legal and all outputs should be maintained.
    - if ***alu_enable_a*** == 0 and ***alu_enable_b*** == 1, it is legal and the ALU should do operation b.
    - if ***alu_enable_a*** == 1 and ***alu_enable_b*** == 0, it is legal and the ALU should do operation a.
    - if ***alu_enable_a*** == 1 and ***alu_enable_b*** == 1, it is illegal and the output depends on RTL that cover this case. (*Note: RTL can have //full case Synopsys so this case will be excluded.)*
- **alu_op_a**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - When it is ***2'b00***, ***alu_out*** should be the ***AND***-ing of the inputs.
  - When it is ***2'b01***, ***alu_out*** should be the ***NAND***-ing of the inputs.
  - When it is ***2'b10***, ***alu_out*** should be the ***OR***-ing of the inputs.
  - When it is ***2'b11***, ***alu_out*** should be the ***XOR***-ing of the inputs.

    (Note: The design may don't have a default case, all cases are covered)
- **alu_op_b**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - When it is ***2'b00***, ***alu_out*** should be the ***XNOR***-ing of the inputs.
  - When it is ***2'b01***, ***alu_out*** should be the ***AND***-ing of the inputs.
  - When it is ***2'b10***, ***alu_out*** should be the ***NOR***-ing of the inputs.
  - When it is ***2'b11***, ***alu_out*** should be the ***OR***-ing of the inputs.

    (Note: The design may don't have a default case, all cases are covered)
- **alu_irq_clr**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - It should be high for at least one clock cycle then it gets de-asserted.
  - It should be high after at least one clock cycle from the assertion of ***alu_irq***.
  - It is an active high clear signal.

- **alu_out**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - Should be as expected, according to the given table, after one clock cycle from applying the inputs.
  - If ***alu_enable*** and ***alu_enable_a*** are asserted, and ***alu_enable b*** is de-asserted while ***alu_op_a*** is ***NAND*** then ***alu_out*** never be ***8'h00***, due to the constrain on ***alu_in_a***.
- **alu_irq**
  - Synchronized to ***alu_clk***, positive edge trigger.
  - It should be asserted after one clock cycle from the event that triggers it.
  - Events that assert it to high, in case of other events ***alu_irq*** maintains its value.
    - When ***alu_enable*** && ***alu_enable***_a && !***alu_enable b***, operation a:
      - If ***alu_op_a*** is AND, ***alu_in_a*** is 8'hFF and ***alu_in_b*** is 8'hFF.
      - If ***alu_op_a*** is NAND and **~*(in_a & in b)*** equals 8'h00.
        (Note: this case can't happen)
      - If ***alu_op_a*** is OR and ***(alu_in_a | alu_in_b)*** equals 8'hE8.
      - If ***alu_op_a*** is XOR and ***(alu_in_a ^ alu_in_b)*** equals 8'h83.
    - When ***alu_enable*** && ***alu_enable_b*** && !***alu_enable a***, operation b:
      - If ***alu_op_b*** is XNOR and ***(alu_in_a ~^ alu_in_b)*** equals 8'hF1.
      - If ***alu_op_b*** is AND and ***(alu_in_a & alu_in b)*** equals 8'hF4.
      - If ***alu_op_b is*** NOR and **~*(alu_in_a | alu_in_b)*** equals 8'hF5.
      - If ***alu_op_b*** is OR and ***(alu_in_a | alu_in_b)*** equals 8'hFF.
  - It should be high until ***alu_irq_clr*** is asserted.
  - It should be de-asserted after one clock cycle from asserting ***alu_irq_clr***.
  - Event has a priority over ***alu_irq_clr,*** when there is an event from the previous ones and ***alu_irq_clr*** is asserted at that time, then ***alu_irq*** should be high.