

Nombre: Iván Reñison

### Ejercicio 1

- a) Dibujar el circuito resultante de la síntesis del siguiente código en SystemVerilog:

```

module circuito
    (input logic  clk, r, enable, sel,
     input logic [7:0] A, B,
     output logic [7:0] Q,
     output logic [3:0] L);
logic [7:0] Q_temp;
always_ff @ (posedge clk, negedge r)
begin
    if (~r)
        Q_temp <= 8'b0;
    else if (enable)
        if (sel)
            Q_temp <= Q_temp ^ A;
        else
            Q_temp <= Q_temp & B;
end
assign L = A[7:4];
assign Q = Q_temp;
endmodule

```

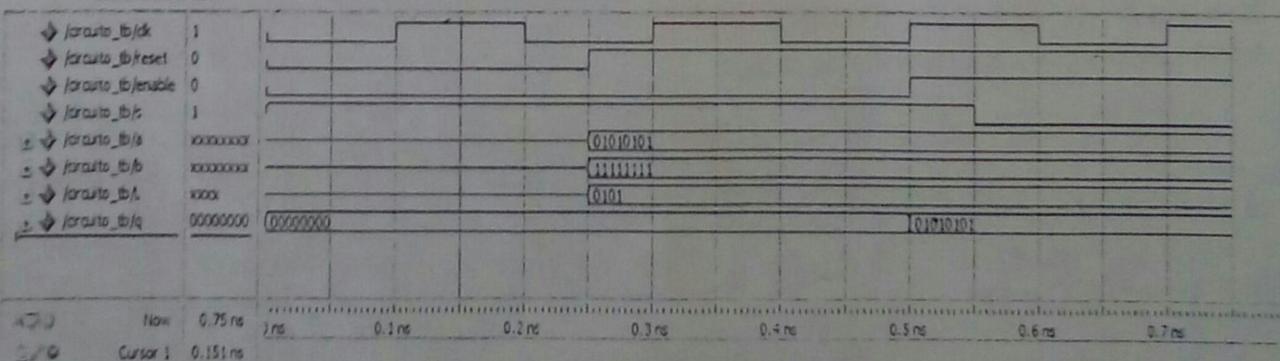
Resuelto al final de todo

- b) A continuación se describe en SystemVerilog el test bench del módulo "circuito". Completar las líneas con los elementos faltantes para obtener las formas de onda de respuesta que se muestran en la figura.

```

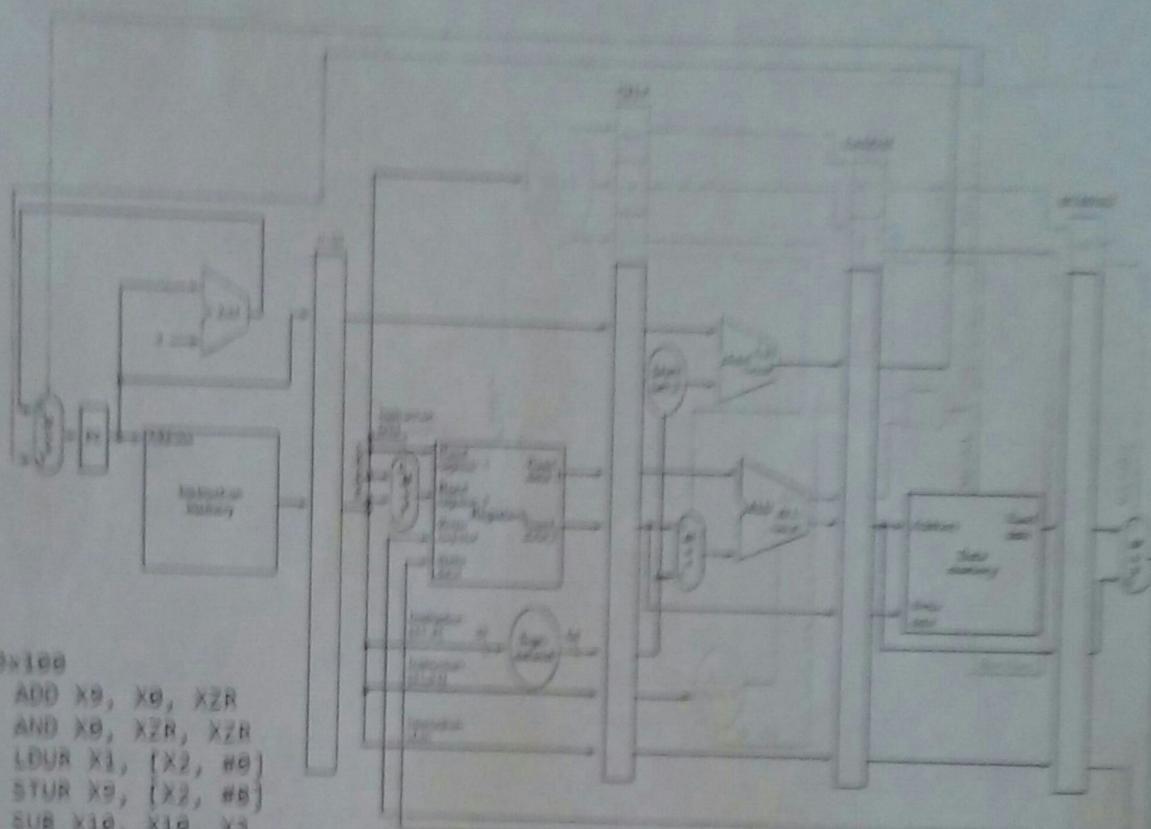
module circuito_tb();
    logic [5:0] clk, reset, enable, s;
    Logic [5:0] L;
    logic [7:0] a, b, q;
    dut (clk, reset, enable, s, a, b, q) L;
    always
        begin
            clk = 0; #100; clk = 1; #100;
        end
    initial
        begin
            reset = 0; enable = 0; s = 1, #0.25ns;
            reset = 1;
            a = 8'b01010101;
            b = 8'b11111111; #0.25ns;
            enable = 1; #0.05ns;
            s = 0; #0.25ns;
            $stop;
        end
endmodule

```



## ✓ Ejercicio 2

El segmento de código dado se ejecuta en el procesador LEGo6 con pipeline de la figura (sin forwarding stall). Analizando el estado del procesador en el ciclo de clock número 5 (considerando que el fetch de la primera instrucción se realiza en el ciclo número 1), responder:



```

.org 0x100
    ADD X9, X9, X2R
    AND X9, X2R, X2R
loop: LDUR X1, [X2, #8]
      STUR X9, [X2, #8]
      SUB X10, X10, X3
      CBZ X3, loop
done: ...

```

- ✓ a) ¿Qué valor hay a la salida del PC? 0x111
- ✓ b) ¿Qué valor hay en la entrada "Write register" del bloque Registers? 0x9
- ✓ c) ¿Qué operación realiza la ALU? sumar el valor de X2 mas 0
- ✓ d) Completar el siguiente cuadro con los valores que tienen las señales de control:

Registros	ALUOp	MemToReg	RegWrite	MemRead	MemWrite	Branch
1	1	0	1	0	0	0

## ✓ Ejercicio 3

A partir de la siguiente secuencia de código:

- 1) LDUR X9, [X1, #8]
- 2) ADDI X9, X9, #3
- 3) STUR X9, [X1, #8]

Tipo de dependencia (registro)	Instrucciones
Dependencia (X9)	1-2
Dependencia (X9)	2-3

**Ejercicio 5**

Considera una sección de código en el vector de excepciones de un microprocesador LEON3 (full ISA), similar al implementado en el Práctico 3. Esta rutina debe:

En el caso de una excepción por interrupción externa, ejecutar un procedimiento alojado en la dirección 0xA4FF0000, usando X0 como argumento que contenga la dirección de retorno de la excepción.

En el caso de una excepción por OpCode invalido, se deberá reemplazar dicha instrucción invalida por la instrucción "ADD Xzr, X0, X0" y retornar de la excepción normalmente.

En cualquier otro caso, retornar de la excepción normalmente sin realizar acción alguna.

- A) Completar el código con las líneas o argumentos faltantes.

```

ESR_VECTOR:
    MRS X9, S2_0_C2_C0_0
    SUBIS XZR, X9, #1
    B.NE ELSE
        MRS X0, S2_0_C0_C0_0
        MOVK X10, #0b10001011000_00000, LSL #16
        BR X10
    ELSE
        SUBIS XZR, X9, #2
        B.NZ ESR_END
        MRS X0, S2_0_C1_C0_0
        MOVK X10, #0b10001011000_00000, LSL #16
        MOVK X10, #0X001F, LSL #0
        STURW X10, [X0, #0]
    ESR_END:
        ERET

```

- B) Seleccionar la respuesta correcta de las siguientes afirmaciones:

1. "Si suponemos que la posición de memoria con la instrucción de label "tricky" está corrompida en memoria en forma permanente (generando un OpCode invalido), considerando la implementación particular de nuestro procesador, el mismo..."

- ... genera un comportamiento impredecible, porque este caso no está contemplado en la lógica de excepciones.
- ... no realiza ninguna acción porque ya está en el vector de excepciones y retorna normalmente.
- ... ante cualquier excepción queda atrapado en el bucle infinito.
- ... ante una excepción por OpCode invalido queda atrapado en el bucle infinito.
- Ninguna es correcta.

2. "Ante la ocurrencia de una excepción no contemplada, este código..."

- ... retorna a la dirección de memoria de la instrucción que generó la excepción.
- ... retorna a la dirección de memoria de la instrucción que generó la excepción + 4.
- ... no puede determinar la dirección de retorno.
- Ninguna es correcta.

Nombre: Tomás Kellman

- ✓ a) Completar la tabla con las dependencias (en la página anterior).  
 b) Mostrar el orden de ejecución considerando un procesador que cuenta únicamente con stall (sin forwarding).  
 c) ¿Cuántos ciclos de reloj tarda el procesador en ejecutar el código del punto b? 11

**Ejercicio 4**

En la siguiente secuencia de código el registro X3 se encuentra inicializado con el valor 10, además, antes de la ejecución del código la posición de memoria apuntada por x5 tiene el valor 13.

- 1> E3: ADDI X1, XZR, #1  
 2> LDUR X2, [X5, #0]  
 3> AND X0, X1, X2  
 4> CBZ X0, E1  
 5> SUBI X2, X2, #1  
 6> B E2  
 7> E1: SUBI X3, X3, #1  
 8> E2: CBNZ X3, E3  
 9> STUR X3, [X5,#0]

Tipo de dependencia (registro)	Instrucciones
Datos	1-3
Datos	2-3
Datos	3-4
Control	4
Datos cond X2	2-5
Datos cond X3	4-5
Datos (condicional) X3	4-7
Control	5
Datos (condicional) X3	4-7

- ✓ a) Completar la tabla con las dependencias.  
 b) Mostrar el orden de ejecución de una iteración del lazo (hasta que se termine de ejecutar la instrucción de salto), considerando un procesador LEGv8 (similar al mostrado en el diagrama del ejercicio 2) pero con forwarding-stall. Dejar correctamente indicados los caminos de forwarding.  
 c) ¿Cuántos ciclos de reloj tarda el procesador en ejecutar el código del punto b? 15  
 ✓ d) Considerando que las etapas del procesador tiene las siguientes latencias:

Instruction Fetch	Instruction Decode	Execute	Memory	Write back
100 ns	60 ns	80 ns	120 ns	70 ns

Completar la siguiente tabla:

	Sin pipeline	Con pipeline
Frecuencia del reloj mínima	430 ns	120 ns
Tiempo que tarda en ejecutar el código del punto b	3010 ns	1800 ns
Mejora de rendimiento al ejecutar el código del punto b	1.692222	