

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica
IE-0424 LABORATORIO DE CIRCUITOS DIGITALES I
II ciclo 2014

Experimento I
MiniAlu Xilinx

Natalia Araya Campos, B00448
Alejandro Leon Torres, B13645
Andres Mora Zuñiga, B14463
Kenneth Vallecillo González, B16750
Grupo 1

Profesor: Diego Valverde G

26 de agosto de 2014

Índice

Índice de figuras

1. Ejercicio I

Para implementar la multiplicación se debe agregar en el archivo *Defintions.v* la siguiente línea:

```
'define MUL    4'b1000
```

Y en el archivo *MiniAlu.v* se debe agregar y alterar el código, se muestra a continuación los cambios más significativos en el archivo:

```
// Se cambia de reg a
// reg wired como lo indica el enunciado.
reg signed [15:0] rResult;

// Se crea una variable temporal para almacenar el resultado de la
// multiplicacion y después tomar la parte más significativa y ponerla
// en rResult.
reg signed [31:0] rResultTemp;

// Se cambia el tipo de wire en un wired signed como lo indica el enunciado
// esto con el fin de obtener un resultado correcto en la multiplicación.
wire signed [15:0] wSourceData0,wSourceData1;

//-----
'MUL:
begin
rFFLedEN    <= 1'b0;
rBranchTaken <= 1'b0;
rWriteEnable <= 1'b1;
rResultTemp <= wSourceData0 * wSourceData1;
rResult    <= rResultTemp [31:16];
end
//-----
```

2. Ejercicio IV

Se tienen que arreglar los warnings que se dan con la compilación de los archivos .v, los mismos se muestran en la Figura ??.

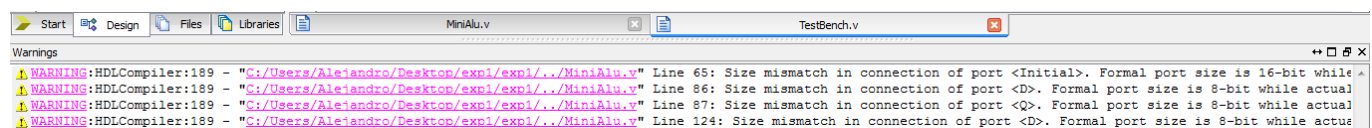


Figura 1: Simulación del pipeline de MiniAlu.

Note que la descripción de todos los warnings se refiere a que el tamaño de los parámetros enviados a alguno de los puertos de las instancias de contadores o flip flops no corresponde al solicitado por la instancia. Así se modificó el código en donde el compilador lo indica, a continuación se muestran las variaciones realizadas:

■ Reparación del warning 1:

• Código original:

```
assign wIPInitialValue = (Reset) ? 8'b0 : wDestination;
UPCOUNTER_POSEDGE IP
(
  .Clock(    Clock                ),
  .Reset(    Reset | rBranchTaken ),
  .Initial(  wIPInitialValue + 1   ),
  .Enable(   1'b1                  ),
  .Q(        wIP_temp              )
);
```

• Código reparado:

```
UPCOUNTER_POSEDGE IP
(
  .Clock(    Clock                ),
  .Reset(    Reset | rBranchTaken ),
  // Se cambia la forma de realizar la suma para evitar el warning.
  .Initial(  wIPInitialValue+16'b1),
  .Enable(   1'b1                  ),
  .Q(        wIP_temp              )
);
```

■ Reparación de los warnings 2 y 3:

• Código original:

```
FFD_POSEDGE_SYNCHRONOUS_RESET # ( 8 ) FFD1
(
  .Clock(Clock),
  .Reset(Reset),
  .Enable(1'b1),
  .D(wInstruction[27:24]),
  .Q(wOperation)
);
```

• Código reparado:

```
// Se cambia el tamaño del parametro para que sea de 4 bits en vez de 8
// siendo 4 el tamaño necesario y eliminando así 2 warnings.
FFD_POSEDGE_SYNCHRONOUS_RESET # ( 4 ) FFD1
```

```
(
  .Clock(Clock),
  .Reset(Reset),
  .Enable(1'b1),
  .D(wInstruction[27:24]),
  .Q(wOperation)
);
```

■ Reparación del warning 4:

● Código original:

```
FFD_POSEDGE_SYNCRONOUS_RESET # ( 8 ) FF_LEDS
(
  .Clock(Clock),
  .Reset(Reset),
  .Enable( rFFLedEN ),
  .D( wSourceData1 ),
  .Q( oLed      )
);
```

● Código reparado:

```
FFD_POSEDGE_SYNCRONOUS_RESET # ( 8 ) FF_LEDS
(
  .Clock(Clock),
  .Reset(Reset),
  .Enable( rFFLedEN ),
  // Se toman solo los primeros 8 bits de wSourceData1 ya que se usan solo 8 LEDs
  // en la tarjeta, note que el cable oLed es de 8 bits también. Así se evita un
  // warning.
  .D( wSourceData1 [7:0] ),
  .Q( oLed      )
);
```

Al final se volvió a compilar este código y el mismo no retornó ningún otro warning.