ELEX 2117 : Digital Techniques 2
2022 Winter Term

Solutions to Midterm Exam 2

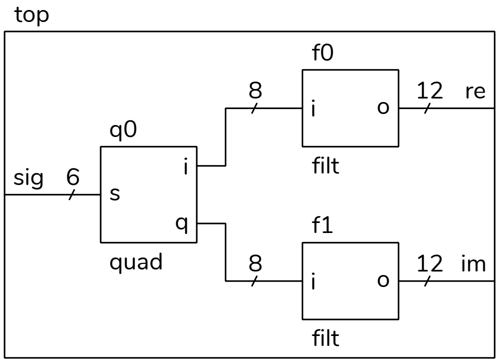
Question 1

The following Verilog code shows the declaration of **quad** and **filt** modules.

```
module quad
  #(n=8)
   ( input  logic [n-1:0] s,
     output logic [  7:0] i, q ) ;
   // ...
endmodule

module filt
  #(m=12)
   ( input  logic [  7:0] i,
     output logic [m-1:0] o ) ;
   // ...
endmodule
```

The~~y~~ diagram shows how they are connected within a **top** module:



The numbers above the signals show the signal (bus) widths. The identifiers above the boxes show the instance names. The identifiers below the boxes show the module names. The identifiers inside the boxes show the port names.

Write a System Verilog module named **top** that implements the diagram above. Declare any signals required to implement this module. Follow the course coding conventions.

Solution

The **top** module needs to be declared with a 6-bit input **sig** and two 12-bit outputs **re** and **im**.

Since the width of the input to the **quad** module is 6 and the default value is 8 (the **parameter** keyword is optional), an explicit value for **n** is required when instantiating the **quad** module. The default value of the **filt o** output is 12 so the default value can be used.

Two 8-bit signals need to be declared to connect the **q0 i** output to the **f0 i** input and to connect the **q0 q** output to the **f1 i** input.

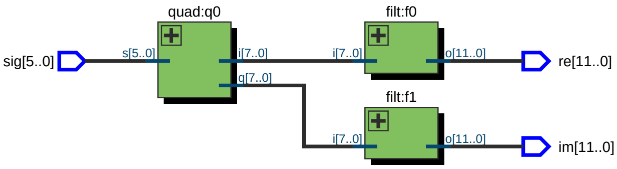
The following Verilog uses explicit names and values for all parameters and explicit names for all ports but parameter names can be omitted and the port names can be omitted if the signal names are specified in the correct order:

```
module top
  ( input  logic [ 5:0] sig,
    output logic [11:0] re, im ) ;

   logic [7:0] a, b ;

   quad #(.n( 6)) q0 ( .s(sig), .i(a), .q(b) ) ;
   filt #(.m(12)) f0 ( .i(a), .o(re) ) ;
   filt #(.m(12)) f1 ( .i(b), .o(im) ) ;
endmodule
```

It synthesizes to the following block diagram:



Question 2

Write [a] System Verilog declaration for a two-dimensional array that models a lookup table of five rows, each containing a 6-bit value equal to the square of the row index. The array should be declared with row indices increasing from 0 to 4 and with decreasing bit indices. For example, the first value in the table would be 6'd0 and the last would be 6'b1000[0]. You may use packed or unpacked arrays and any valid literal format.

Solution

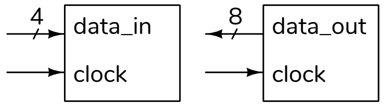
The first dimension specifies the allowed row indices (0 to 4) and the second specifies the allowed bit indices for a 6-bit value (5 to 0). The array is initialized with values from $0^2 = 0$ to $4^2 = 16$ using literals in any base. For convenience decimal base is used below. Any name could be used for the table. The width and base of the constants (`6'd`) could also be omitted since the values would be truncated to the required width.

```
logic [0:4][5:0] squares =
    '{ 6'd0, 6'd1, 6'd4, 6'd9, 6'd16 } ;
```

Question 3

The following diagram shows the signals making up an interface. The number of bits for each signal are shown above the signal, the arrow shows the port type (input or output) and the name of each signal describes its function.

There were two versions of the question:



Answer the following questions about this interface. Briefly (in one sentence or less) justify your answer.

- is it serial or parallel?
- is it synchronous or asynchronous?
- is it uni-directional or bi-directional?

Solution

- This is a parallel interface because more than one bit is being transferred at the same time.
- This is a synchronous interface because it uses a clock signal
- This is a uni-directional interface because data is only transferred in one direction (in or out).

Question 4

Fill in the blank boxes in the table below so that all values in each row are consistent (agree with each other). The first row is an example.

There were two versions of the question:

signal name	truth value (T/F)	logic level (H/L)	Verilog value as output (0/1)	Verilog value in expression (0/1)
$\overline{\texttt{hot}}$	T	L	0	1
alarm*	F	$\boxed{\text{H}}$	$\boxed{1}$	$\boxed{0}$
even	$\boxed{\text{F}}$	L	$\boxed{0}$	$\boxed{0}$
$\overline{\texttt{link}}$	$\boxed{\text{F}}$	$\boxed{\text{H}}$	1	$\boxed{0}$

and:

signal name	truth value (T/F)	logic level (H/L)	Verilog value as output (0/1)	Verilog value in expression (0/1)
$\overline{\text{hot}}$	T	L	0	1
alarm	T	$\boxed{\text{H}}$	$\boxed{1}$	$\boxed{1}$
even*	$\boxed{\text{T}}$	L	$\boxed{0}$	$\boxed{1}$
link	$\boxed{\text{T}}$	$\boxed{\text{H}}$	1	$\boxed{1}$

Solution

The truth value and the logic level for active-low (shown by an overbar or trailing (*)) and active-high signal names are shown below:

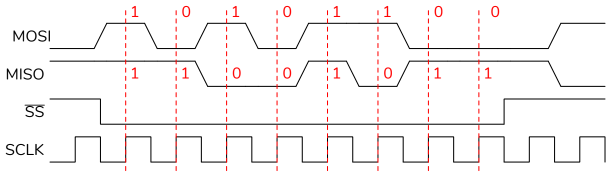
signal name	truth value (T/F)	logic level (H/L)	Verilog I/O	Verilog expression value
$\overline{\texttt{s}}$	T	L	0	1
$\overline{\texttt{s}}$	F	H	1	0
s	F	L	0	0
s	T	H	1	1

and this table can be used to fill in the truth value, logic level and "Verilog Value in Expression" columns in the table by using the values in the corresponding row.

Verilog always uses the active-high convention for inputs and outputs: `0` and `1` are treated as low and high output levels respectively and this can be used to fill in the "Verilog value as output" or the logic level columns.

The answers are shown in boxes in the tables above.

Question 5



The waveform[s] above is[are] measured on an SPI interface. What value was transferred from the master to the slave (or slave to the master)? Give your answer as a Verilog constant with the correct width and using a hexadecimal base. Show how you obtained your answer. Assume the bits are transferred most-significant-bit first.

Solution

The **`MOSI`** and **`MISO`** signals are (typically) sampled on the rising edge of **`SCLK`** but only while $\overline{\text{SS}}$ is asserted (low). These values are shown in the diagram above. The bits transferred from master to slave, on **`MOSI`**, are **`8'b1010_1100`** = **`8'hac`** and from slave to master, on **`MISO`**, are **`8'b1100_1011`** = **`8'hcb`**.