

Assignment 4

Dears,

In this assignment we will practice on Behavioral level. In Behavioral Modeling it's not necessary to put the schematic!

- Use Non-Blocking statement in behavioral model.
- All projects must contain of Test fixture.
- Pay attention to deadline!

Good Luck!

Computer Architecture Lab Team

Behavioral Modeling:

1. **Divide Algorithm** (Section 3.4 of Computer Architecture Book):

The necessary flowcharts are given here too. You can see one example of this algorithm in Fig 3.11 of computer Architecture book.

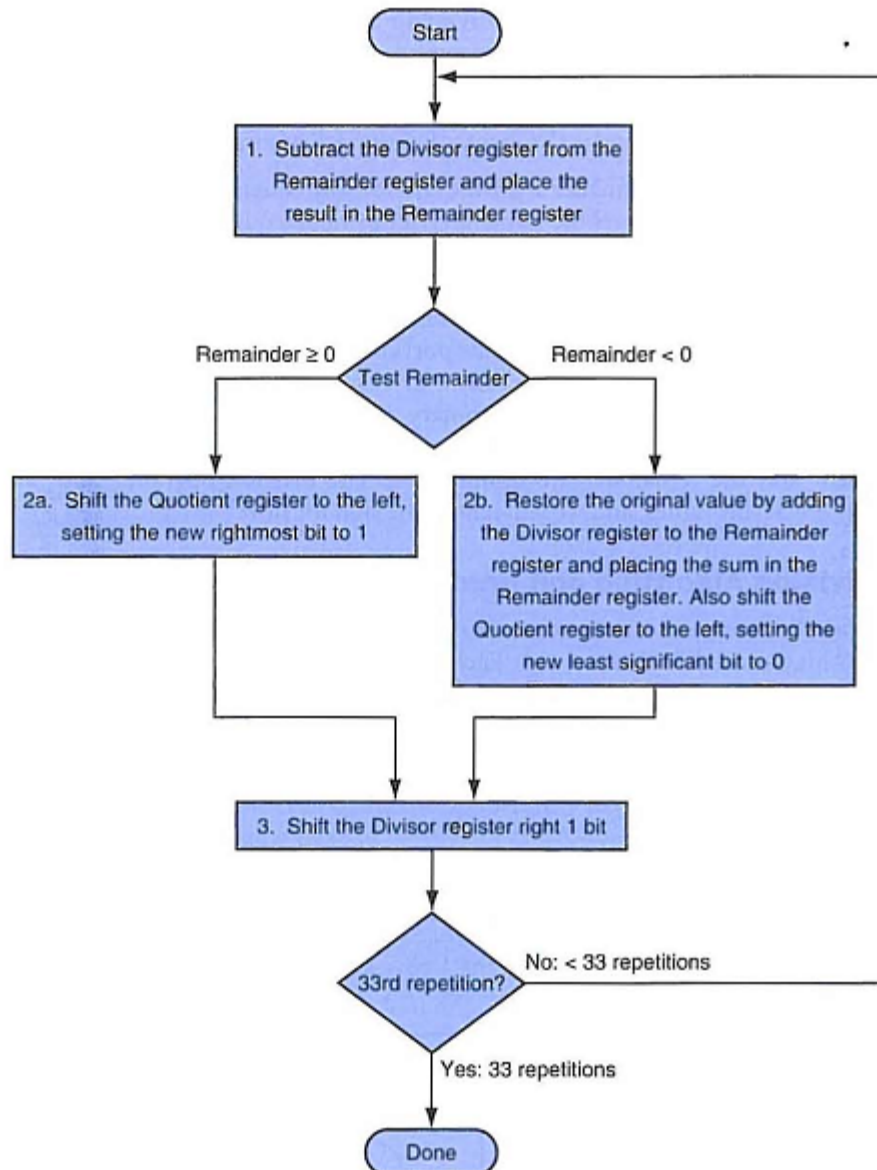


FIGURE 3.10 A division algorithm, using the hardware in Figure 3.9. If the remainder is positive, the divisor did go into the dividend, so step 2a generates a 1 in the quotient. A negative remainder after step 1 means that the divisor did not go into the dividend, so step 2b generates a 0 in the quotient and adds the divisor to the remainder, thereby reversing the subtraction of step 1. The final shift, in step 3, aligns the divisor properly, relative to the dividend for the next iteration. These steps are repeated 33 times.

Iteration	Step	Quotient	Divisor	Remainder
0	Initial values	0000	0010 0000	0000 0111
1	1: Rem = Rem - Div	0000	0010 0000	0110 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0010 0000	0000 0111
	3: Shift Div right	0000	0001 0000	0000 0111
2	1: Rem = Rem - Div	0000	0001 0000	0111 0111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0001 0000	0000 0111
	3: Shift Div right	0000	0000 1000	0000 0111
3	1: Rem = Rem - Div	0000	0000 1000	0111 1111
	2b: Rem < 0 \Rightarrow +Div, sll Q, Q0 = 0	0000	0000 1000	0000 0111
	3: Shift Div right	0000	0000 0100	0000 0111
4	1: Rem = Rem - Div	0000	0000 0100	0000 0011
	2a: Rem \geq 0 \Rightarrow sll Q, Q0 = 1	0001	0000 0100	0000 0011
	3: Shift Div right	0001	0000 0010	0000 0011
5	1: Rem = Rem - Div	0001	0000 0010	0000 0001
	2a: Rem \geq 0 \Rightarrow sll Q, Q0 = 1	0011	0000 0010	0000 0001
	3: Shift Div right	0011	0000 0001	0000 0001

FIGURE 3.11 Division example using the algorithm in Figure 3.10. The bit examined to determine the next step is circled in color.

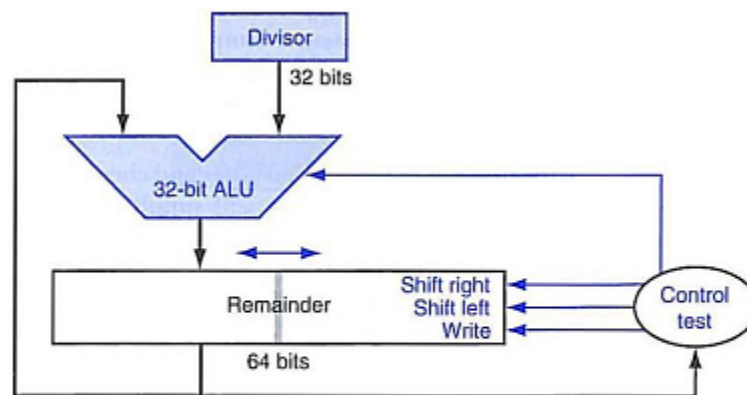


FIGURE 3.12 An improved version of the division hardware. The Divisor register, ALU, and Quotient register are all 32 bits wide, with only the Remainder register left at 64 bits. Compared to Figure 3.9, the ALU and Divisor registers are halved and the remainder is shifted left. This version also combines the Quotient register with the right half of the Remainder register. (As in Figure 3.6, the Remainder register should really be 65 bits to make sure the carry out of the adder is not lost.)

Hint3: You can use always #10000 clk = ~clk; in test code to generate 10Hz clock cycle.

And change the timescale for the delay is defined by the **`timescale** statement at the beginning of the file to **"`timescale 1ms / 1ms"**.

2. How to study

Now, you have learnt behavioral modeling! So you can model many things in this type of modeling. Consider below question and write your idea about it in Verilog language at behavioral model. (The pseudo code is enough. Syntax is not important; just clarify your understanding about this module in text file not in Xilinx!)

The name of this module is: How should I study in BIHE University?

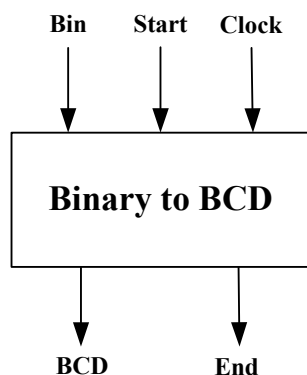
1. What are the inputs and outputs of this module?
2. What are sensitivity list of always block from *your* view?
3. What should sensitivity list of always block be from *Bahai* view?
4. How can you test this module?

- If you have some problems in understanding the questions don't hesitate to ask your question.

3. You should implement binary to BCD. This module is very useful especially in driving 7 segment. We will use this module in future to see your project result on 7 segment in Spartan board.

۲. مبدل دودویی به دهدهی

مشخصات مدار مورد نظر بصورت زیر است:



ورودی:

- عدد دودویی (۳۲ بیت) Bin :
- شروع عملیات (۱ بیت) Start :
- Clock

خروجی:

- عدد دهدهی (۳۲ بیت) BCD :
- پایان عملیات (۱ بیت) End :

با فعال شدن سیگنال Start مدار شروع به کار کرده و ورودی Bin را به معادل دهدهی آن تبدیل کرده و حاصل را روی خطوط BCD می گذارد و سیگنال End را به منزله اعلام پایان عملیات فعال می کند.

توجه : تمام ورودی و خروجی ها باید با Clock سنکرون باشند.

الگوریتم تبدیل یک عدد دودویی به دهدهی معادل به صورت زیر است :

- الف : عدد دودویی ورودی را یک بیت به چپ شیفت دهید.
- ب : اگر رقم \bar{A} م عدد دهدهی بزرگتر یا برابر ۵ است به آن رقم، ۳ تا اضافه کنید.
- ج : مراحل الف و ب را به اندازه تعداد بیت های عدد دودویی تکرار کنید.

مثال: تبدیل عدد دودویی ۱۱۰۱۰۰۰۱۰۰ به معادل دهدهی

عمل	عدد دودویی	رقم ۱	رقم ۲	رقم ۳
شیفت به چپ	001101000100	0000	0000	0000
شیفت به چپ	011010001000	0000	0000	0000
شیفت به چپ	110100010000	0000	0000	0000
شیفت به چپ	101000100000	0001	0000	0000
شیفت به چپ	010001000000	0011	0000	0000
به رقم ۱ سه تا اضافه کن	100010000000	0110	0000	0000
شیفت به چپ	100010000000	1001	0000	0000
شیفت به چپ	000100000000	0011	0001	0000
به رقم ۱ سه تا اضافه کن	001000000000	0110	0010	0000
شیفت به چپ	001000000000	1001	0010	0000
به رقم ۲ سه تا اضافه کن	010000000000	0010	0101	0000
شیفت به چپ	010000000000	0010	1000	0000
شیفت به چپ	100000000000	0100	0000	0001
به رقم ۱ سه تا اضافه کن	000000000000	1001	0000	0010
شیفت به چپ	000000000000	1100	0000	0010
به رقم ۱ سه تا اضافه کن	000000000000	1000	0001	0100
شیفت به چپ	000000000000	1011	0001	0100
پایان عملیات	000000000000	0110	0011	1000

خروجی دهمی معادل

عدد دودویی ورودی