3.9 3.10 3.11 3.13 3.16 3.18

In 3.18, "Assume both inputs are unsigned 6-bit integers" should be "Assume both inputs are unsigned 8-bit integers"

3.9 [10] <§3.2> Assume 151 and 214 are signed 8-bit decimal integers stored in two's complement format. Calculate 151 + 214 using saturating arithmetic. The result should be written in decimal. Show your work.

151
214

$(151)_{10} = (10010111)_2 \rightarrow (01101001)_2 = -(105)$

$(214)_{10} = (11010110)_2 \rightarrow (00101010)_2 = (-42)$

$$\begin{array}{r} 10010111 \\ 11010110 \\ \hline 01101101 \end{array}$$

$151 + 214 =$

$$\begin{array}{r} 01101001 \\ 00101010 \\ \hline 10010011 \end{array} = (-147)_{10}$$

$-105-42$

because of saturating arithmetic, the result is (-128)

$\boxed{10010011}$

3.10 [10] <§3.2> Assume 151 and 214 are signed 8-bit decimal integers stored in two's complement format. Calculate 151 − 214 using saturating arithmetic. The result should be written in decimal. Show your work.

-105

$-105-(-42)$

$(151)_{10} = (10010111)_2 \rightarrow (01101001)_2$

$(214)_{10} = (11010110)_2 \rightarrow (00101010)_2$

$$\begin{array}{r} 10010111 \\ 00101010 \\ \hline 11000001 \end{array}$$

$151 - 214 =$

$$\begin{array}{r} 01101001 \\ 00101010 \\ \hline (00111111)_2 \end{array} = (63)_{10} < 128$$

So the result is 63.

3.11 [10] <§3.2> Assume 151 and 214 are unsigned 8-bit integers. Calculate 151 + 214 using saturating arithmetic. The result should be written in decimal. Show your work.

$(151)_{10} = (10010111)_2$

$(214)_{10} = (11010110)_2$

$151 + 214 = \quad 10010111$

11010110

$$\underbrace{(101101101)_2}> (255)_{10}$$

since we use saturating arithmetic, so the result is $(255)_{10}$.

3.13 [20] <§3.3> Using a table similar to that shown in Figure 3.6, calculate the product of the hexadecimal unsigned 8-bit integers 62 and 12 using the hardware described in Figure 3.5. You should show the contents of each register on each step.

$(62)_{16} = (01100010)_2$

$(12)_{16} = (00010010)_2$

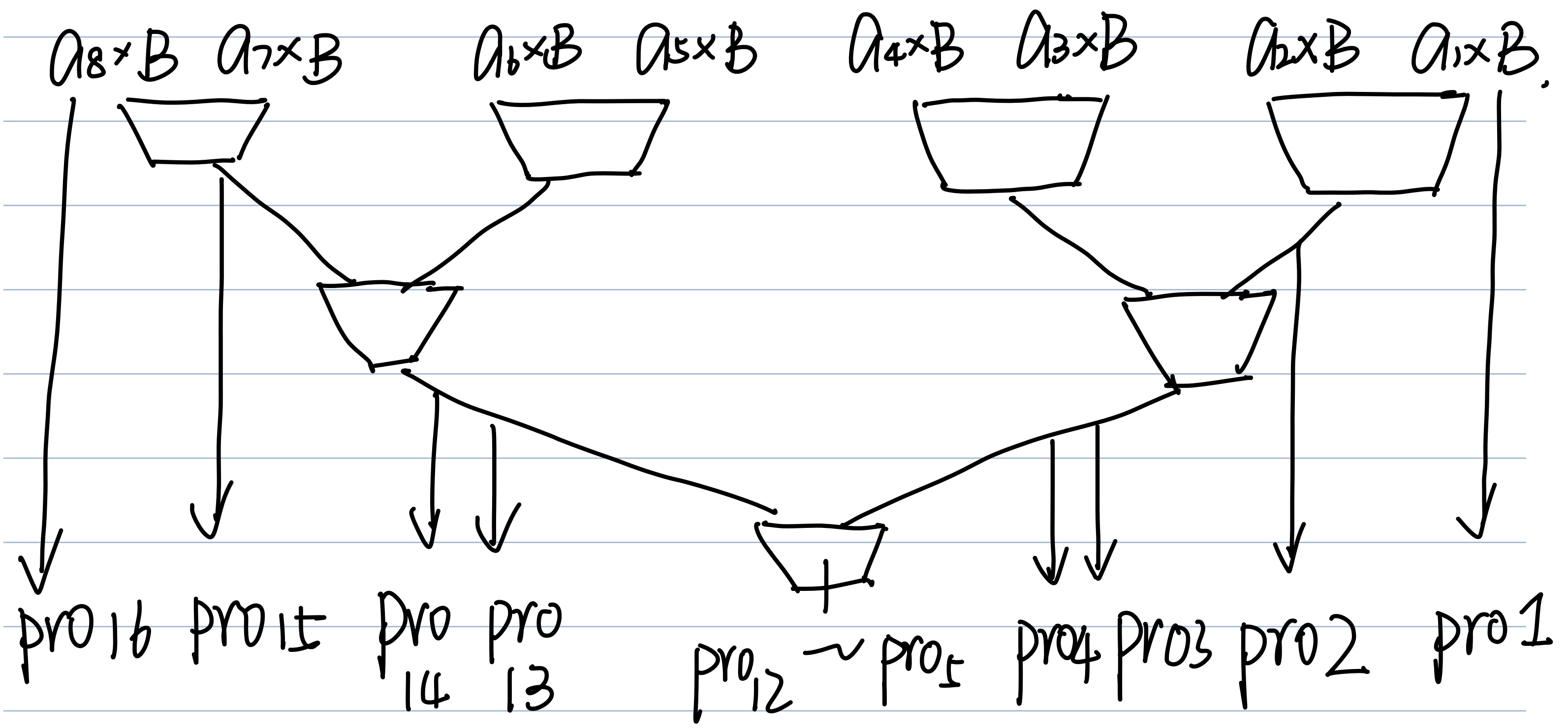
content of register

Iteration	Step	product (15 bits)	multiplicand
0	initial	00000000 ¦ 01100010	00010010
1	0: no operation		00010010
	shift pro right	000000000 ¦ 0110001	00010010
2	1: add multiplicant to left bit of product	000100100 ¦ 0110001	00010010
	shift pro pright	0000100100 ¦ 011000	00010010
3	0: no operation		00010010
	shift pro right	00000100100 ¦ 01100	00010010
4	0: no operation		00010010
	shift pro right	000000100100 ¦ 0110	00010010
5	0: no operation		00010010
	shift pro right	0000000100100 ¦ 011	00010010
6	1: add	0001001100100 ¦ 011	00010010
	shift pro right	00001001100100 ¦ 01	00010010

7	1: add	00011011100100⁝01	00010010
	shift pro right	00001101110010⁝00	00010010
8	0: no operation		00010010
	shift pro right	0000011011100100	00010010

So the result is → 0000011011100100 = $(06E4)_{16}$.

3.16 [20] <§3.3> Calculate the time necessary to perform a multiply using the approach given in Figure 3.7 if an integer is 8 bits wide and an adder takes 4 time units.



So the time nessary is 12 time units.

3.18 [20] <§3.4> Using a table similar to that shown in Figure 3.10, calculate 74 divided by 21 using the hardware described in Figure 3.8. You should show the contents of each register on each step. Assume both inputs are unsigned 8-bit integers. 大卬除小卬

$(74)_{10} = (01001010)_2$ $(21)_{10} = (00010101)_2$

contents of register

Iteration	Step	quotient	Divisor	Remainder
0	initial values	0000 0000	0001 0101 0000 0000	0000 0000 0100 1010
1	1: Rem = Rem − Div	0000 0000	0001 0101 0000 0000	1110 1011 0100 1010
	Rem<0, +Div, sll Q, $Q_0=0$	0000 0000	0001 0101 0000 0000	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 1010 1000 0000	0000 0000 0100 1010
2.	Rem = Rem − Div	0000 0000	0000 1010 1000 0000	1111 0101 1100 1010
	Rem<0, +Div, sll Q, $Q_0=0$	0000 0000	0000 1010 1000 0000	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 0101 0100 0000	0000 0000 0100 1010
3	Rem −= Div	0000 0000	0000 0101 0100 0000	1111 1011 0000 1010
	Rem<0, +Div, sll Q, $Q_0=0$	0000000	0000 0101 0100 0000	0000 0000 0100 1010
	shift Divisor right	0000000	0000 0010 1010 0000	0000 0000 0100 1010
4	Rem −= Div	0000 0000	0000 0010 1010 0000	1111 1101 1010 1010
	Rem<0, +Div, sll Q, $Q_0=0$	0000 0000	0000 0010 1010 0000	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 0001 0101 0000	0000 0000 0100 1010
5	Rem −= Div	0000 0000	0000 0001 0101 0000	1111 1110 1111 1010
	Rem<0 +Div sll Q $Q_0=0$	0000 0000	0000 0001 0101 0000	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 0000 1010 1000	0000 0000 0100 1010
6	Rem −= Div	0000 0000	0000 0000 1010 1000	1111 1111 1010 0010
	Rem<0 +Div sll Q $Q_0=0$	0000 0000	0000 0000 1010 1000	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 0000 0101 0100	0000 0000 0100 1010
7	Rem −= Div	0000 0000	0000 0000 0101 0100	1111 1111 1111 0110
	Rem<0 +Div sll Q $Q_0=0$	0000 0000	0000 0000 0101 0100	0000 0000 0100 1010
	shift Divisor right	0000 0000	0000 0000 0010 1010	0000 0000 0100 1010
8	Rem −= Div	0000 0000	0000 0000 0010 1010	0000 0000 0010 0000
	Rem>0 sll Q $Q_0=1$	0000 0001	0000 0000 0010 1010	0000 0000 0010 0000

	shift Divisor right	0000 0001	0000 0000 0001 0101	0000 0000 0010 0000
9	Rem -= Div	0000 0001	0000 0000 0001 0101	0000 0000 0000 1011
	Rem>0 sll Q $Q_0=1$	0000 0011	0000 0000 0001 0101	0000 0000 0000 1011
	shift Divisor right	0000 0011	0000 0000 0000 1010	0000 0000 0000 1011

So the quotient is $(11)_2$ and the remainder is $(1011)_2$.