CSCI 113 Assignment 6

1. Slt instruction is done by performing A-B and taking the resulting sign bit as the result. If A<B then sign bit will be 1 and if A>=B then sign bit will be 0.

The control signal is 0111 because the control signal is a combination of bit inputs to the ALU. The signal is represented by A inv, B inv, S0 and S1 respectively. S0 + S1 are the input bits that control the Mux that selects the output. For slt A inv is set to 0 and B inv is set to 1 to allow for A-B to be done. The sign bit from A-B is chosen by setting S0 & S1 to 1. That's why the control signal for slt is 0111.

2. 0101 * 1110   5 * 14 = 70

MD	AC	MQ	Step	iterations
0101	0000	1110		0
	0000	1110	AC+0	1
	0000	0111	>>	1
	0101	0111	AC+MD	2
	0010	1011	>>	2
	0111	1011	Ac+MD	3
	0011	1101	>>	3
	1000	1101	AC+MD	4
	0100	0110	>>	4

result = 0100 0110 = 70 ✓

3a) 1010011101

place	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Booth's val	$\bar{1}$	1	$\bar{1}$	0	1	0	0	$\bar{1}$	1	$\bar{1}$
decimal	-512	256	-128	0	32	0	0	-4	2	-1

sum of decimal values: $-512 + 256 - 128 + 32 - 4 + 2 - 1$
$= -355$

proof: convert original value to decimal

$1010011101 \xrightarrow{2's\ comp} -\ 01\ 0110\ 0011$

$-(0 + 256 + 0 + 64 + 32 + 0 + 0 + 0 + 2 + 1)$
$= -355$

b) 1010 * 0101 $-6 * 5 = -30$

MD	AC	MQ	MQ-1	Step	iterations
1010	0000	0101	0		0
1010	0110	0101	0	AC-MD	1
1010	0011	0010	1	>>	1
1010	1101	0010	1	AC+MD	2
1010	1110	1001	0	>>	2
1010	0100	1001	0	AC-MD	3
1010	0010	0100	1	>>	3
1010	1100	0100	1	AC+MD	4
1010	1110	0010	0	>>	4

answer: 1110 0010 → -30 ✓

4 1011 / 0100   11/4 = 2   remainder:3

MD	AC	MQ	Step	iteration
0100	0000	1011		0
0100	0001	0110	<<	1
0100	1101	0110	AC-MD	1
0100	1101	0110	MQ0=0	1
0100	0001	0110	restore	1
0100	0010	1100	<<	2
0100	1110	1100	AC-MD	2
0100	1110	1100	MQ0=0	2
0100	0010	1100	restore	2
0100	0101	1000	<<	3
0100	0001	1000	AC-MD	3
0100	0001	1001	MQ0=1	3
0100	0011	0010	<<	4
0100	1111	0010	AC-MD	4
0100	1111	0010	MQ0=0	4
0100	0011	0010	restore	4

MQ = Q = 0010 = 2 ✓

AC = R = 0011 = 3 ✓

5) 1011 / 0100   11/4 = 2 remainder 3

MD	AC	MQ	Step	iteration
0100	0000	1011		0
0100	0001	0110	<<	1
0100	1101	0110	AC-MD	1
0100	1101	0110	MQ0=0	1
0100	1010	1100	<<	2
0100	1110	1100	AC+MD	2
0100	1110	1100	MQ0=0	2
0100	1101	1000	<<	3
0100	0001	1000	AC+MD	3
0100	0001	1001	MQ0=1	3
0100	0011	0010	<<	4
0100	1111	0010	AC-MD	4
0100	1111	0010	MQ0=0	4
0100	0011	0010	AC+MD	DONE

MQ = Q = 0010 = 2 ✓

Ac = R = 0011 = 3 ✓

6, Restoring division does 3*B+R operations, where B is # of bits and R is the # of restores done at most equal to B.

Non-restoring division does 3*B+F operations, where B is # of bits and F is number of times the final operation runs (at least once).

They both achieve the same performance in the best case, but in the worst case the non-restoring division is much faster because it doesn't have to do multiple extra operations like the restoring division.