Booth's Algorithm Exercise

Uses Registers as follows

- M: multiplicand
- M-: 2s complement of M
- Q: multiplier
- Q_{-1}: one bit register to the right of Q, padded with 0
- A: accumulator or result, initially 0
- A/Q/Q_{-1} treated as a single shift register during shift operations
- At each stage, action defined by the Q_0Q_{-1} transition
- 1-0: A ← A – M
- 0-1: A ← A + M
- At each step, A/Q/Q_{-1} is shifted right with sign extension

Example: 3 X 7, 4 bit inputs, 8 bit result

7 (M)
X 3 (Q)

A	Q	Q_{-1}	M	M-	Action
0000	0011	0	0111	1001	Initial
1001	0011	0			1-0 transition → A = A - M
1100	1001	1			Shift A/Q/Q_{-1} right one bit (sign extend on left)
1100	1001	1			1-1 transition → no op, A unchanged
1110	0100	1			Shift A/Q/Q_{-1}
0101	0100	1			0-1 transition → A = A + M
0010	1010	0			Shift A/Q/Q_{-1}
0010	1010	0			0-0 transition → no op, A unchanged
0001	0101	0			Shift A/Q/Q_{-1}

Result is 00010101 = 16 + 4 + 1 = 21

Example: 3 X -7, 4 bit inputs, 8 bit result

-7 (M)
X 3 (Q)

A	Q	Q_{-1}	M	M-	Action
0000	0011	0	1001	0111	Initial
0111	0011	0			1-0 transition → A = A - M
0011	1001	1			Shift A/Q/Q_{-1} right one bit (sign extend on left)
0011	1001	1			1-1 transition → no op, A unchanged
0001	1100	1			Shift A/Q/Q_{-1}
1010	1100	1			0-1 transition → A = A + M
1101	0110	0			Shift A/Q/Q_{-1}
1101	0110	0			0-0 transition → no op, A unchanged
1110	1011	0			Shift A/Q/Q_{-1}

Result is 11101011 = -00010100+1 = -00010101 = -(16+4+1) = -21

Exercise: 6 X 4

A	Q	Q_{-1}	M	M-	Action
0000		0			Initial
					Action: Shift:
					Action: Shift:
					Action: Shift:
					Action: Shift:

Result is:

Exercise: 6 X -4

A	Q	Q_{-1}	M	M-	Action
0000		0			Initial
					Action: Shift:
					Action: Shift:
					Action: Shift:
					Action: Shift:

Result is: