

8-Bit Multiply Program

This program multiplies two unsigned 8-bit numbers to produce an 8-bit result. The arguments are found in B and C and the result is placed in X. It is expected to take about 3 minutes. Overflow is ignored. It modifies: A, B, C, D, X, Y.

Example input:

B = 0001 0001 = 17
 C = 0000 1101 = 13

Output:

X = 1101 1101 = 221

Example input:

B = 0001 1100 = 28
 C = 0000 1001 = 9

Output:

X = 1111 1100 = 252

Address	Instruction		
0000 0000	0011 1001	Y=B	Y = B
0000 0001	0011 0110	X=0	X = 0
0000 0010	1000 0101	A= \neg B	If sign(Y)==1
0000 0011	1111 0000	BNEG Else	.
0000 0100	0000 0000	.	.
0000 0101	0000 0111	.	.
0000 0110	0011 0010	X=C	X = C
		Else:	.
0000 0111	0101 1001	A=-7	D = -7
0000 1000	0001 1000	D=A	.
		Loop:	Loop:
0000 1001	0000 1110	B=X	Shift X left circular
0000 1010	1000 0110	A=B<<1	.
0000 1011	0011 0000	X=A	.
0000 1100	0000 1111	B=Y	Shift Y left circular
0000 1101	1000 0110	A=B<<1	.
0000 1110	0011 1000	Y=A	.
0000 1111	0000 1111	B=Y	If sign(Y)=1
0001 0000	1000 0101	A= \neg B	.
0001 0001	1111 0000	BNEG Else2	.
0001 0010	0000 0000	.	.

8-Bit Multiply Program

```
0001 0011    0001 0111    .          .
0001 0100    0000 1110    B=X        X = X + C
0001 0101    1000 0000    A=B+C      .
0001 0110    0011 0000    X=A        .
                Else2:    .
0001 0111    0000 1011    B=D        D = D + 1
0001 1000    1000 1001    D=B+1     .
0001 1001    1110 0010    BNZ Loop  if D != 0 goto Loop
0001 1010    0000 0000    .         .
0001 1011    0000 1001    .         .
0001 1100    1010 1110    HALT     HALT
```