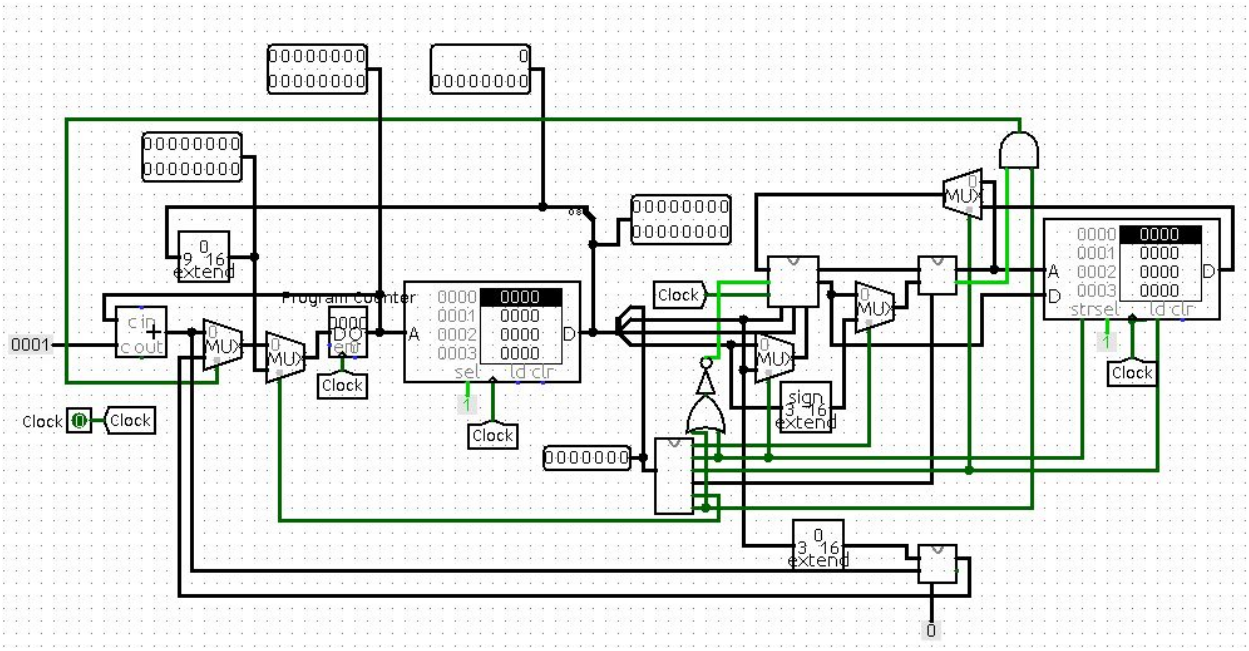
Name: Bobby Headrick

Processor Name: 16 Bit Rock

Processor Description: 16 single cycle CPU with Branch and Jump support

Score Sheet

Feature	Expected Value	Expected Completion
Basic: ALU	10	3/16
Basic: R and I type	10	3/16
Basic: LW and SW	10	3/16
Basic: Assembler	20	3/17
Basic: Writeup	20	3/20
16 Bit	5	3/16
BEQ	10	3/21
CLA	10	3/21
Jump	10	3/21
Total	105	



The processor I developed is a 16 bit single cycle processor which supports basic I and R type instructions, as well as the branch if equal instruction and the jump instruction. The ALU also uses a carry-lookahead adder. The instruction word is 16 bits, with the highest order 7 bits making up the OPCODE. The OPCODE can be broken down (starting from high order) to:

- Immediate
- SW
- LW
- ALU OP (2 bits)
- Jump
- BEQ

R-Type Instructions

OPCODE	Write Address	Read Address 1	Read Address 2
7 bits	3 bits	3 bits	3 bits

I-Type Instructions

OPCODE	Write Address	Read Address 1	Immediate Value
7 bits	3 bits	3 bits	3 bits

LW

OPCODE	Write Address	Read Address 1	Offset
7 bits	3 bits	3 bits	3 bits

SW

OPCODE	Read Address	Write Address	Offset
7 bits	3 bits	3 bits	3 bits

BEQ

OPCODE	Read Address 1	Read Address 2	Offset
7 bits	3 bits	3 bits	3 bits

Jump

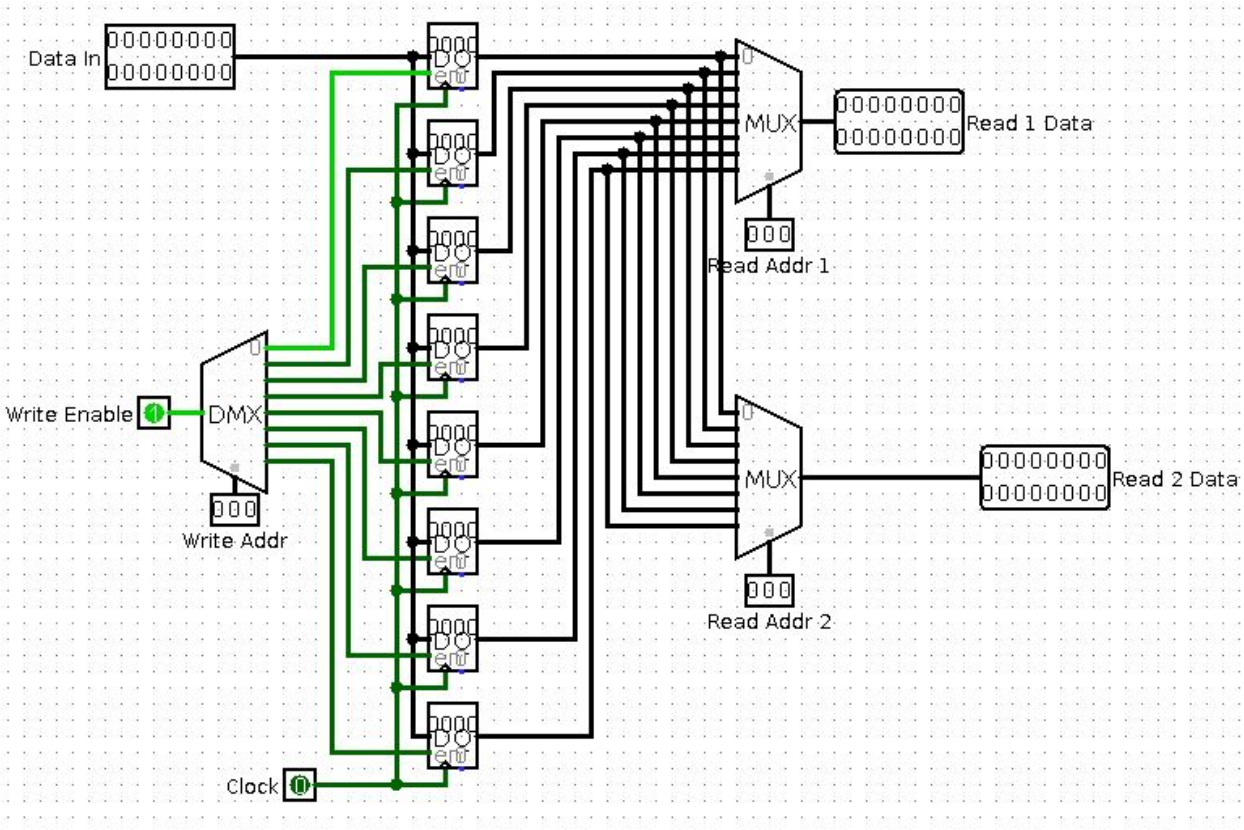
OPCODE	Target
7 bits	9 bits

Directions for Assembler:

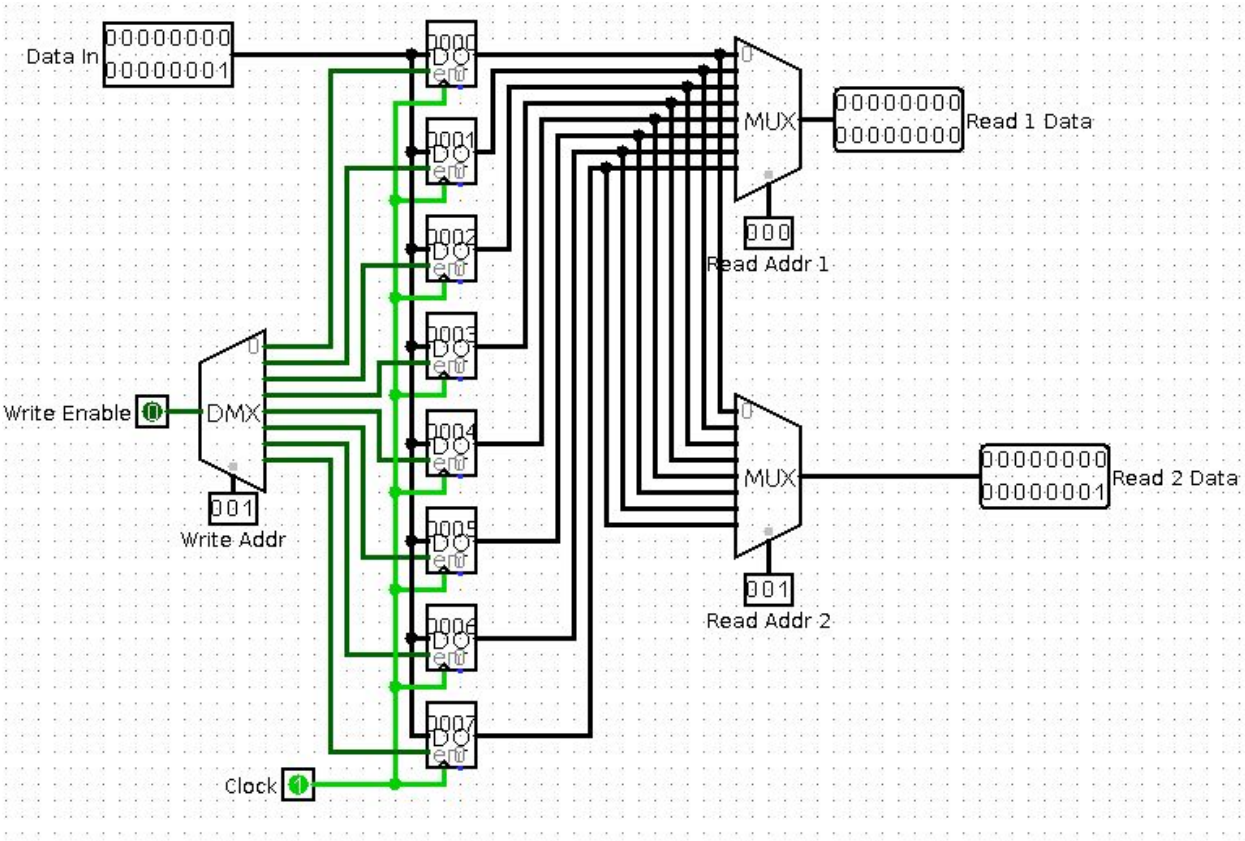
Python skeleton-assembler.py inputfile.asm outputfile.hex

R and I-Type before and after:

Register file before adding values 0-7 to registers $0-$7:



Registers after (uses both add and addi):



SW before and after:

Data memory empty:

0000	0000	0000	0000	0000	0000	0000	0000	0000
0008	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000
0018	0000	0000	0000	0000	0000	0000	0000	0000

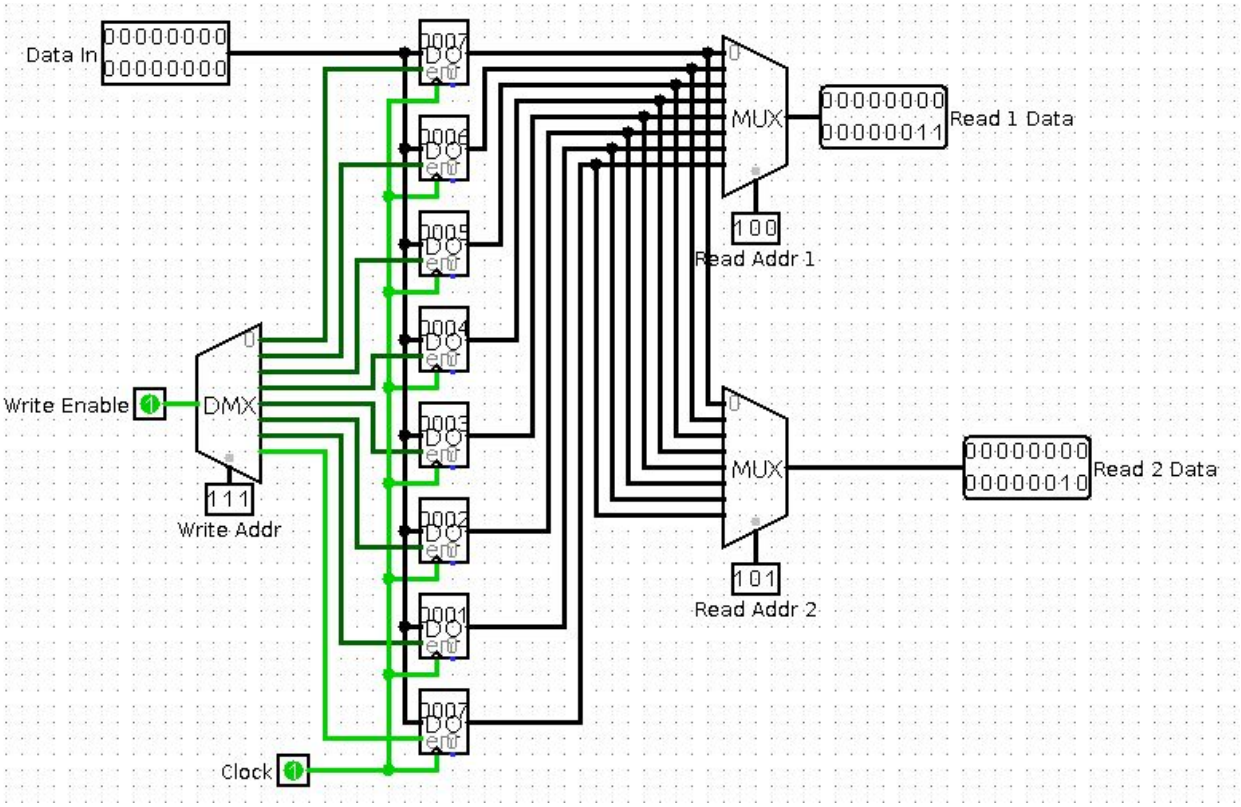
Data memory populated with values in register:

0000	0000	0001	0002	0003	0004	0005	0006	0007
0008	0000	0000	0000	0000	0000	0000	0000	0000
0010	0000	0000	0000	0000	0000	0000	0000	0000
0018	0000	0000	0000	0000	0000	0000	0000	0000

LW before and after:

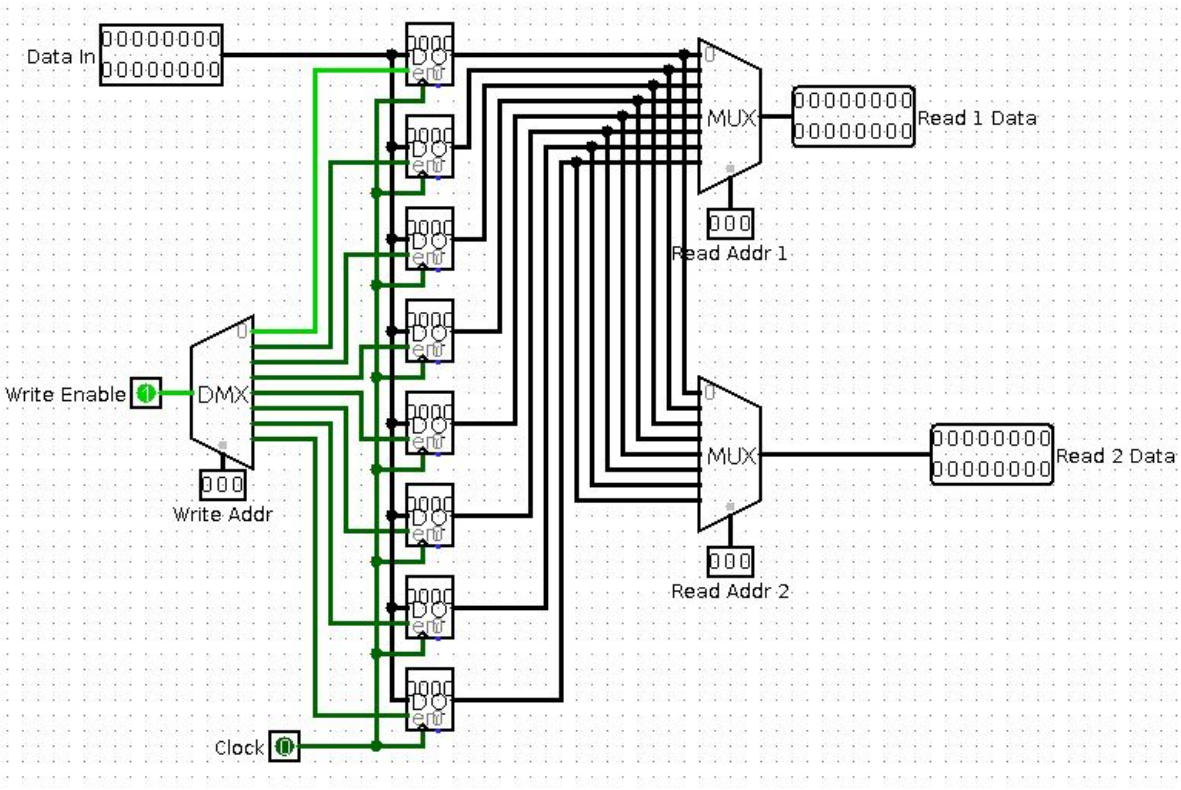
Before: see register after I and R type instructions.

After values in data memory loaded in (part of reverse):

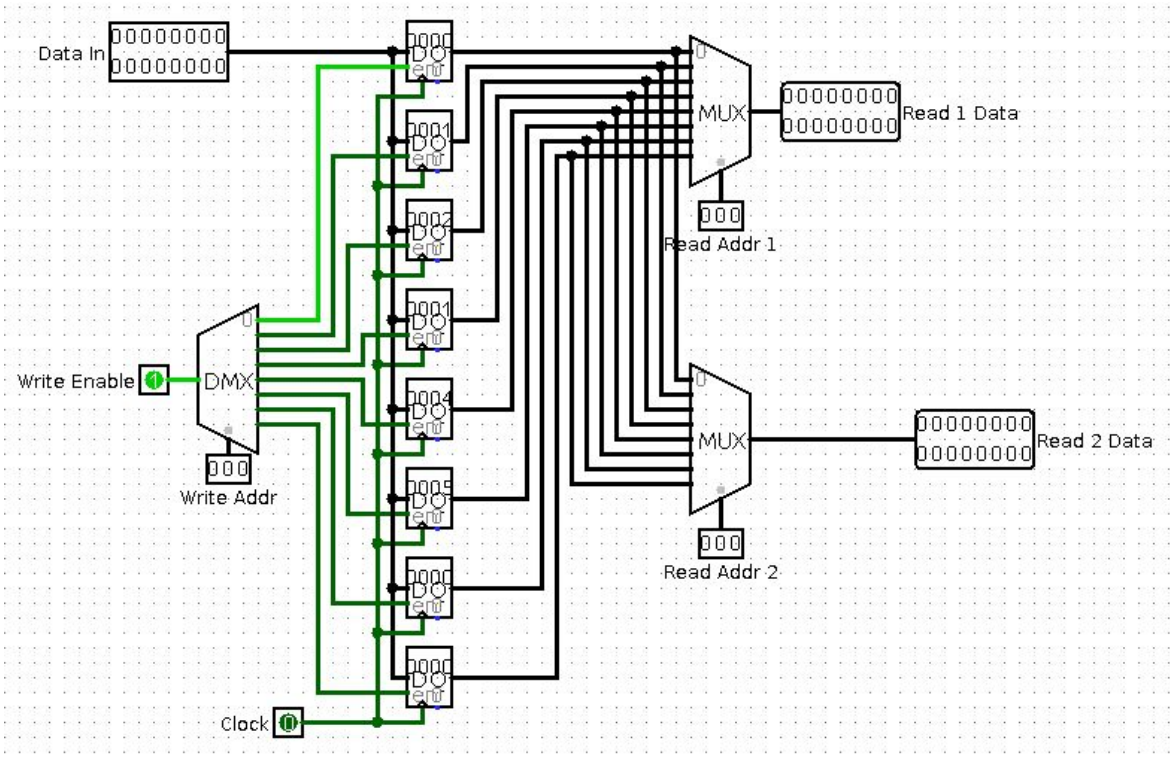


BEQ before and after:

Register File before:

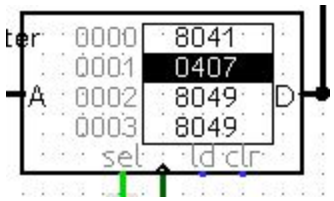


Register File after running Branch Test:

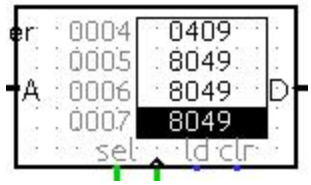


Jump before and after:

Current spot in instruction memory, with jump call to 0007 in 0001:

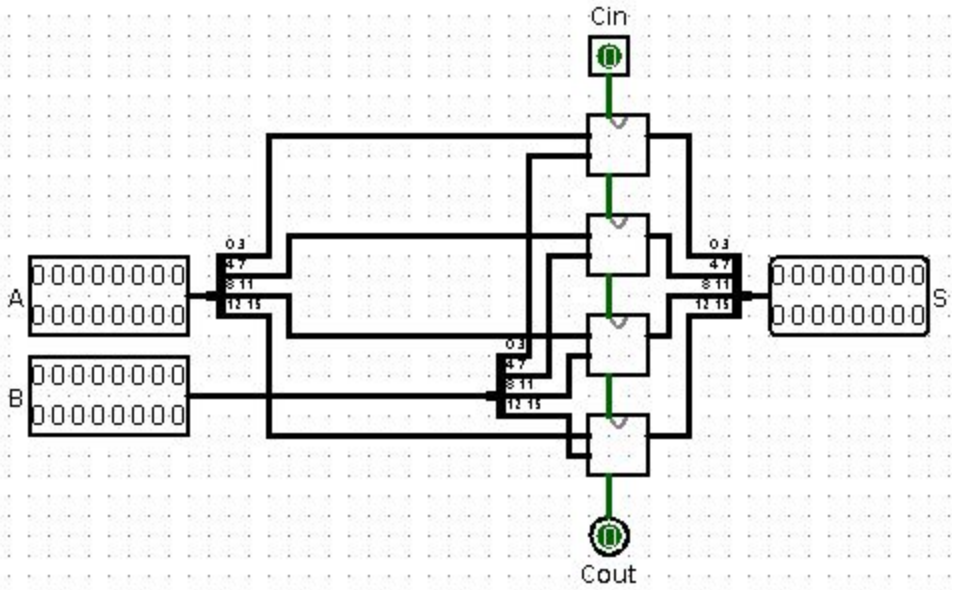


Current spot in instruction memory immediately after one clock cycle:

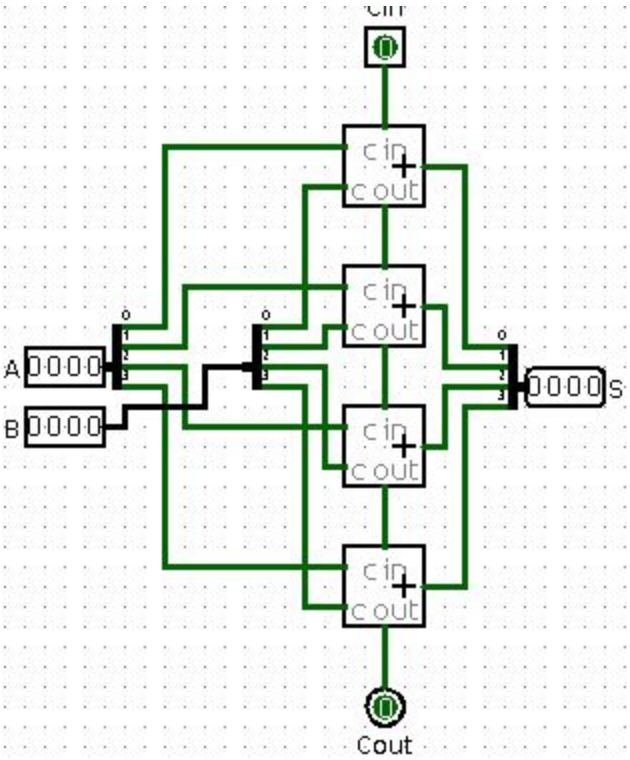


Carry-Lookahead Adder:

16-bit carry-lookahead adder composed of four 4-bit carry-lookahead adder blocks:



Single 4-bit carry-lookahead adder block:


Cin
cin
cout
cin
cout
cin
cout
cin
cout
A
0000
B
0000
0000
S
Cout