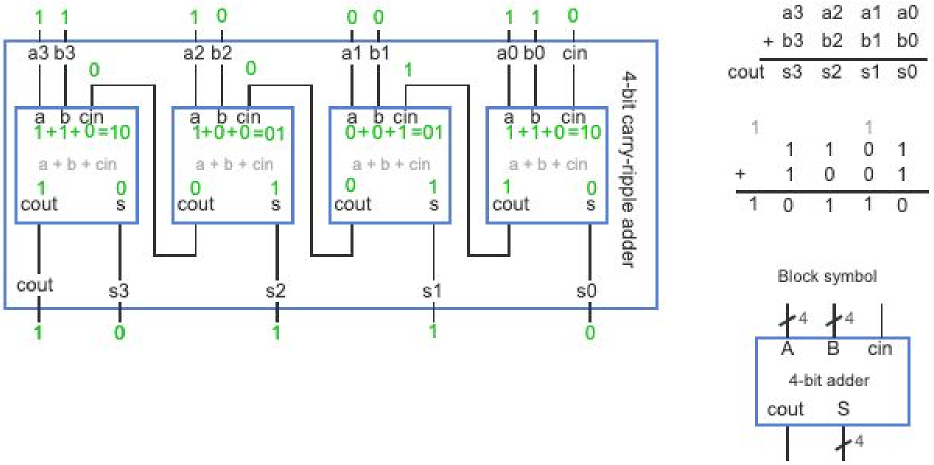
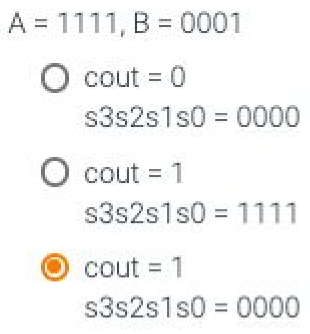
4.1 Adders

- Binary adders, 1+1 carries 1
- Computes A+B
    - A, B are N-bit numbers
    - Carry-ripple adder



    -
        - N-bit carry-ripple adder adds two N-bit numbers
        - N is the size of each input
        - C_{out} is the carry bit, 1 indicates a carry



A = 1111, B = 0001

○ cout = 0
s3s2s1s0 = 0000

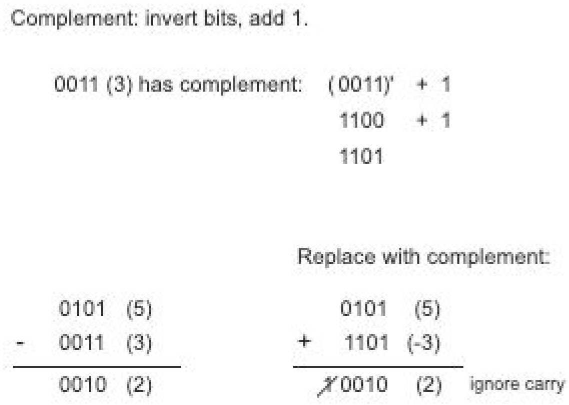
○ cout = 1
s3s2s1s0 = 1111

● cout = 1
s3s2s1s0 = 0000

            -
    - **Full adder**
        - circuit that adds three bits and generates a sum and carry-out.
        - Half adder
            - circuit that adds two bits and generates a sum and carry-out bit
        - An N-bit carry-ripple adder is constructed with N full adders
- **Incrementer**
    - Adds 1 to a number (Adds 1 to input A)
    - Usually use half adder to implement incrementer but Full adder also works, but it requires larger circuits

4.2 Signed numbers in Binary

- Unsigned - positive only
- Signed - both positive and negative
- Use the left bit for the sign, 0 - positive, 1 - negative
- **Two’s complement signed number representation**
    - For negative numbers, calculate its complement
        - Invert 1 and 0
        - Add 1 to the result
    - Add two numbers up, and ignore the carry bit
    - example



Complement: invert bits, add 1.

0011 (3) has complement: (0011)' + 1
1100 + 1
1101

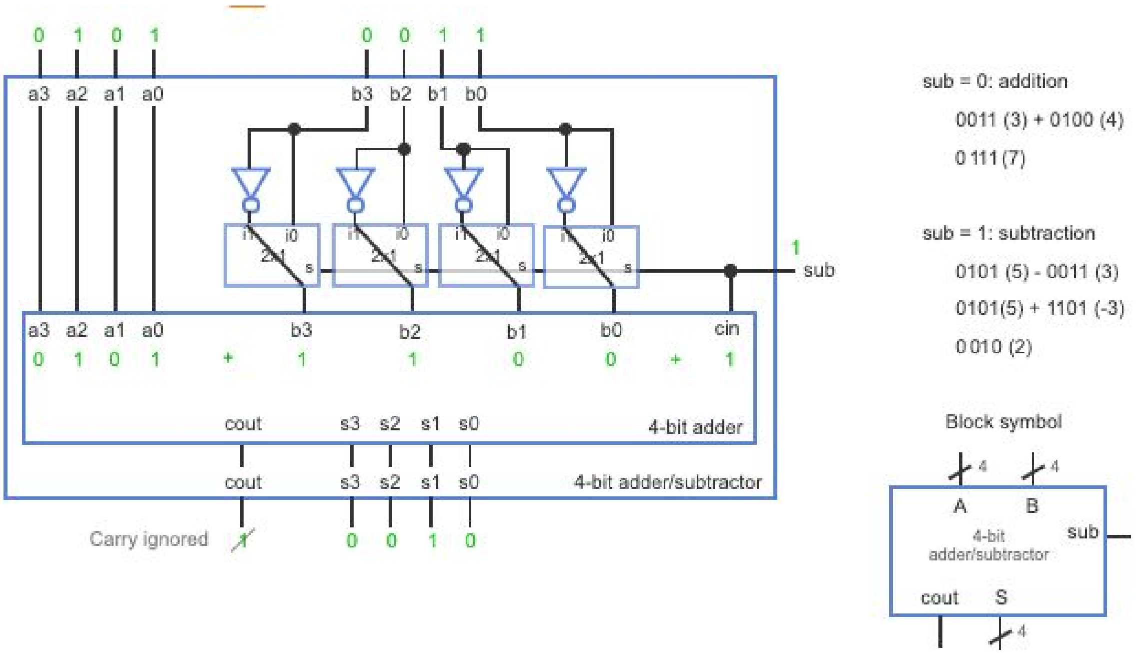
Replace with complement:

	0101	(5)		0101	(5)
-	0011	(3)	+	1101	(-3)
	0010	(2)		~~1~~0010	(2) ignore carry

    -
    - More examples
        - 1011, negative because the left bit is 1
        - 1001, negative because the left bit is 1, in decimal
            - 0110 + 1 = 0111
            - Magnitude is 7, yields to -7 in decimal
        - -3 in eight-bit two’s complement representation is
            - 00000011
            - 11111100 + 1
            - 11111101
    - Allows adder to deal with negative number additions
        - 3 + -4
        - 0011 + 1100
        - 1111 (two’s complement representation)
        - 0000 + 1 = 0001, magnitude is 1, in decimal is -1
- **Overflow**
    - Adding two positives, or adding two negatives, may yield a value that can't be represented in the given number of bits
    - Example
        - 0001 + 1111 = 0000
        - Adding a negative number and a positive number cannot result in overflow
        - 0111 + 0110 = 1101 (-3)
        - 6 + 7 results overflow

4.3 Subtractors

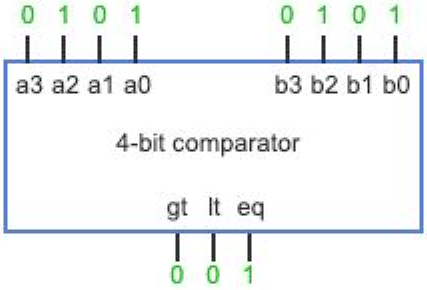
- Computes A - B
- Using two’s complement representation for (-B)
- Because two's-complement representation performs subtraction by complementing and adding, a single adder circuit can perform either addition or subtraction, thus saving circuit size



-
-

4.4 Comparators

- Def: compares two numbers, indicating whether the numbers are equal, or which number is greater.

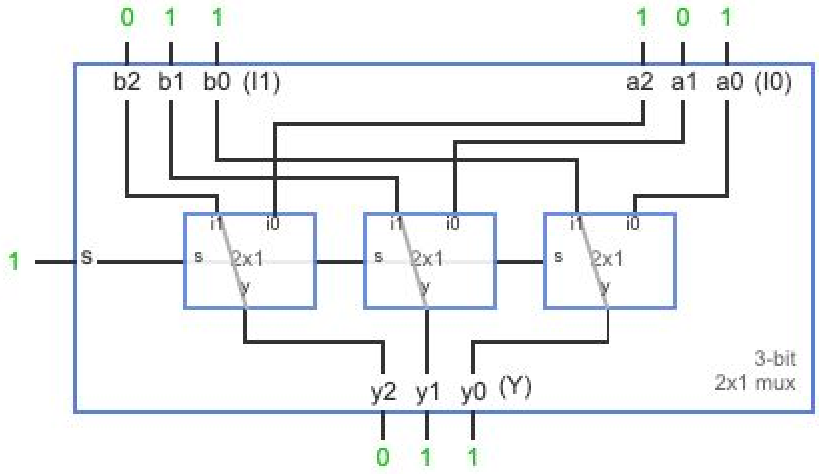


-

- Compare bit by bit
- Carry-ripple comparator
    - compares two N-bit numbers from left to right, with the result of each digit's comparison "rippling" to the next digit

- For each digit, a ***one-bit comparator*** compares two bits a and b only if the eq input was 1 from the higher digit
- else just passing along a gt 1 or an lt 1

4.5 N-bit Muxes

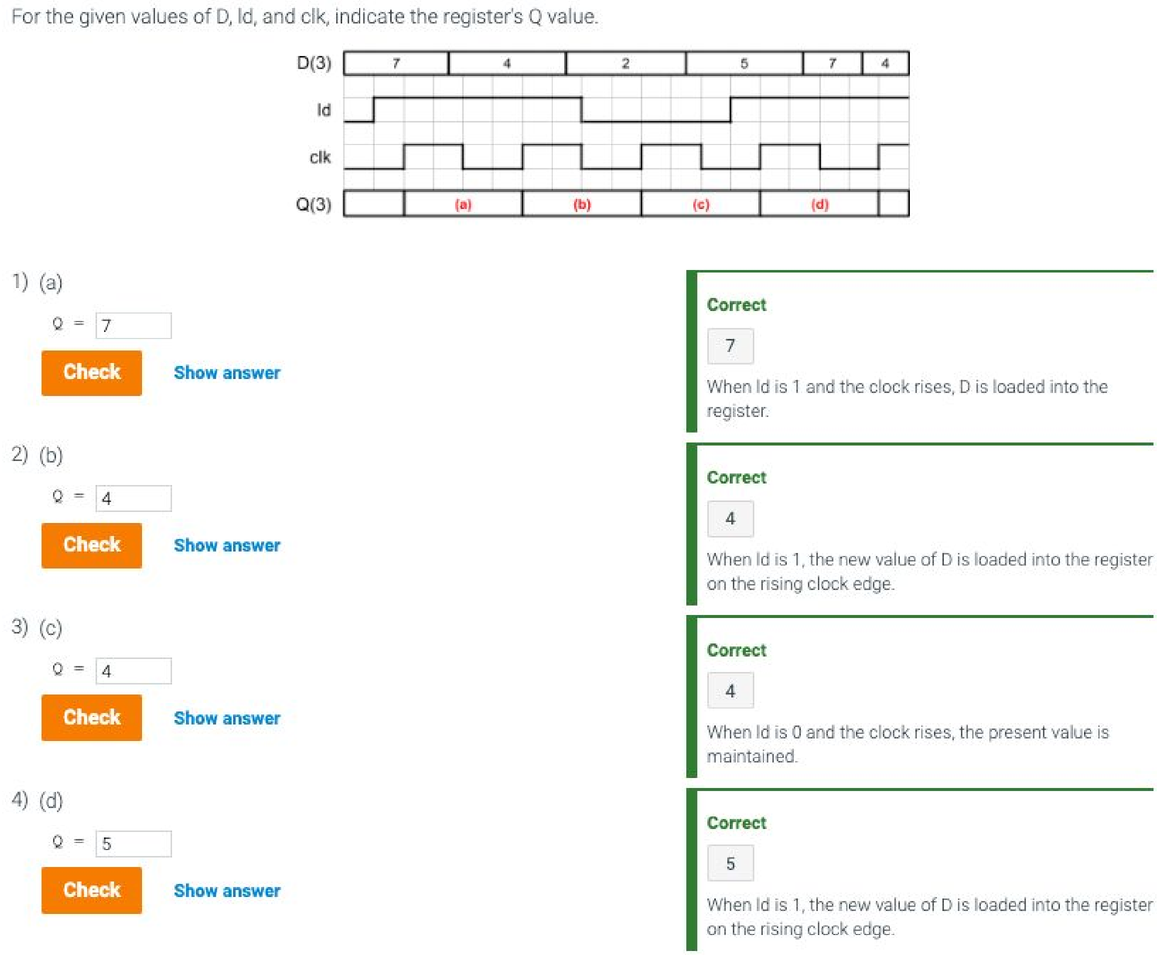


-

- 3-bit 2x1 Mux.

4.6 Load Registers

- Load register design
- Only want to load a register on certain clock cycles, rather than every clock cycle
- Registers come along with a control input “ld” or “load”
- Implementing such a load register can be done using 2x1 muxes



For the given values of D, ld, and clk, indicate the register's Q value.

1) (a)

Q = 7

Check Show answer

Correct

7

When ld is 1 and the clock rises, D is loaded into the register.

2) (b)

Q = 4

Check Show answer

Correct

4

When ld is 1, the new value of D is loaded into the register on the rising clock edge.

3) (c)

Q = 4

Check Show answer

Correct

4

When ld is 0 and the clock rises, the present value is maintained.

4) (d)

Q = 5

Check Show answer

Correct

5

When ld is 1, the new value of D is loaded into the register on the rising clock edge.

-

- An N-bit ***load register*** (such as an 8-bit load register or just 8-bit register) stores N bit values
- Loading new bit values when a clock input rises **if a load input is 1**
- A load input (ld) indicates when the register should be loaded.
- A reset input (rst) may exist that indicates that the register's bits should be reset to 0 (having priority over ld)