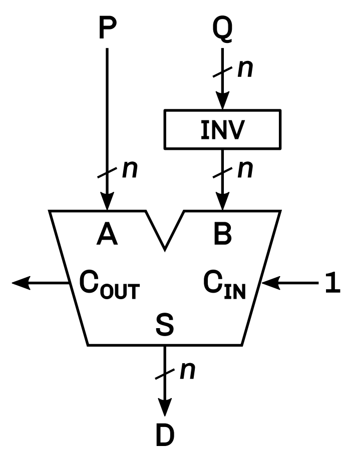
Lecture 15 In-Class Worksheet

1 Learning Objectives

This worksheet is based on Lumetta course notes section 2.5. After completing this lesson, you will know how to:

- [S15-1] Use an adder to build a unsigned and two's complement subtractor circuit.
- [S15-2] Use an adder/subtractor for unsigned and two's complement comparison.

2 Subtractors from Adders

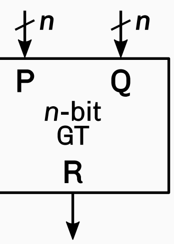


As discussed in the Lumetta course notes (Sec. 2.5.1), we can build a subtractor circuit using a n-bit adder by inverting the subtrahend and providing 1 as the initial carry in, as shown on the right. When operating on integers in unsigned representation, inverting and adding 1 is equivalent to providing $2^n - Q$ as the B input to the adder, giving result $D = P + 2^n - Q = 2^n + P - Q$. If $P < Q$ then $D < 2^n$, so the carry out of the adder will be 0. (Recall that the carry out is just bit n of the result.) On the other hand, if $P \geq Q$, then $D \geq 2^n$, so the carry out will be 1. This means that the carry out bit tells us the result of subtraction would be negative (an overflow condition): if the carry out bit is 0, then the result would be negative, and if the carry out is 1, the result is positive and representable as an unsigned n-bit number.

3 Unsigned Comparison Using Adders

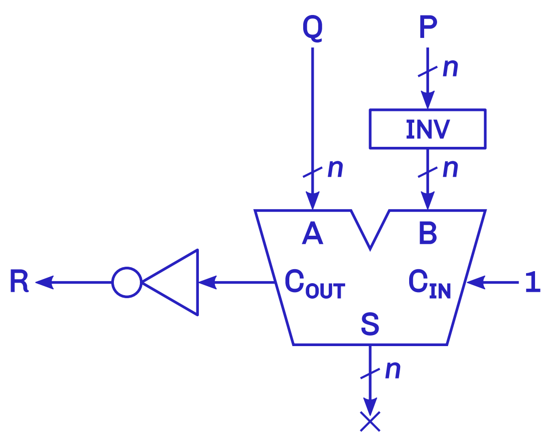
We can use a subtractor to build a comparator. As discussed in the previous section, for two integers in unsigned representation, if $P < Q$, subtracting $P - Q$ as described above will produce a carry out of 0 from the adder, and if $P \geq Q$, then the carry out will be 1. We can use this fact to build an unsigned comparator.

Q1. Build an unsigned comparator using an adder configured for subtraction that behaves like the Unsigned Greater-Than comparator from the Lecture 14 Worksheet, reproduced below.



R	Meaning
0	$P \leq Q$
1	$P > Q$

To test for $P > Q$, we need to subtract $Q - P$. If the carry is 0, then we know that $Q < P$.



Note that if we compute $P - Q$ and use $R = C_{OUT}$, then what we're actually testing is $P \geq Q$.

4 Two's Complement Comparison Using Adders

For integers in two's complement representation, overflow is indicted by:

$$\overline{P_{n-1}}\, Q_{n-1}\, D_{n-1} \;+\; P_{n-1}\, \overline{Q_{n-1}}\, \overline{D_{n-1}}.$$

We can also use an adder configured for subtraction to compare two's complement numbers. Let P and Q are two integers in two's complement representation. What does the result of subtraction $P-Q$ tell us?

- **No overflow.** If there was no overflow, then the sign tells us whether $P-Q$ is negative or not. If $P-Q$ is negative, then $P < Q$. If $P-Q$ is zero or positive (sign bit is zero), then $P \geq Q$.
- **Overflow.** If overflow occurs, then we need to know in which direction the overflow occurred: whether the result was too large in the negative direction ($P-Q < -2^{n-1}$) or in the positive direction ($P-Q \geq 2^{n-1}$). The first case happens when P is negative and Q is positive; the sign bit of the result D is then 0 (the result appears non-negative). The second case happens when P is zero or positive and Q is negative, and the sign bit of the result D is 1 (the result appears negative).

Q2. Derive the logic formula for the condition $P > Q$ as a function of the sign bit N of the result and a signal V that signals two's complement overflow ($V = 1$ means overflow) for the operation $Q - P$.

The truth table based on the signals is given below:

V	N	$P > Q$?
0	0	0
0	1	1
1	0	1
1	1	0

The function above is XOR, so the expression to detect $P > Q$ is $N \oplus V$.

5 General Comparison using Subtraction

It is common for an Arithmetic and Logic Unit (ALU), such as one in a CPU, to produce the following signals along with the result itself:[1]

Sig.	*Meaning*
N	Sign bit of the result
Z	Result is zero (Z=1) or non-zero (Z=0)
V	Two's complement overflow (V=1) or not (V=0)
C	Carry out

Combined with arithmetic operations, these can be used to perform many kinds of comparisons.

Q3. Fill in the table below with an arithmetic operation and a logical expression using the signals N, Z, V, and C to detect the conditions $P = Q$, $P \neq Q$, $P > Q$, $P \geq Q$, $P < Q$, $P \leq Q$ for unsigned and two's complement values.

[1] The ALU in the LC-3 CPU does not do this.

Cond.	*Oper.*	*Unsigned*	*Two's Complement*
$P = Q$	$P - Q$	Z	Z
$P \neq Q$	$P - Q$	$\overline{Z}$	$\overline{Z}$
$P > Q$	$Q - P$	$\overline{C}$	$N \oplus V$
$P \geq Q$	$P - Q$	C	$\overline{N \oplus V}$
$P < Q$	$P - Q$	$\overline{C}$	$N \oplus V$
$P \leq Q$	$Q - P$	C	$\overline{N \oplus V}$