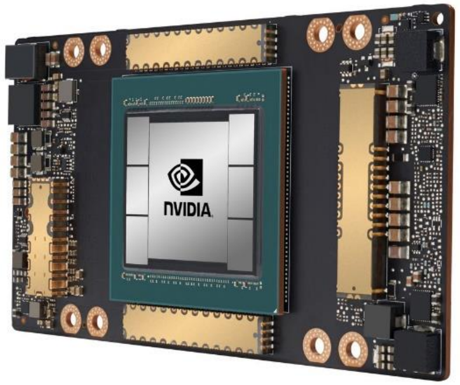
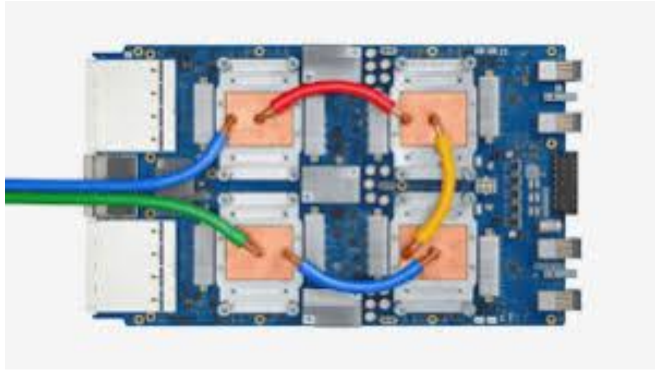
CSCB58: Computer Organization



Prof. Gennady Pekhimenko

University of Toronto

Fall 2020



The content of this lecture is adapted from the lectures of Larry Zheng and Steve Engels

CSCB58 Week 3: Summary

Week 3 review

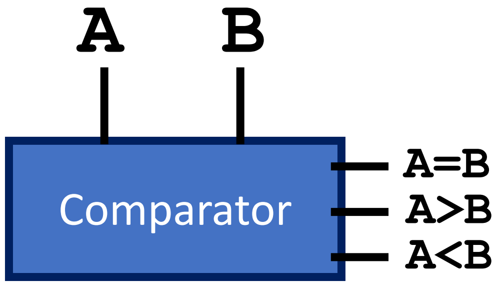
- Building
  - Mux/demux
  - Decoders
  - Adders
  - Subtractors
  - Comparators

Comparators

Comparators

- A circuit that takes in two input vectors, and determines if the first is greater than, less than or equal to the second.
- How does one make that in a circuit?

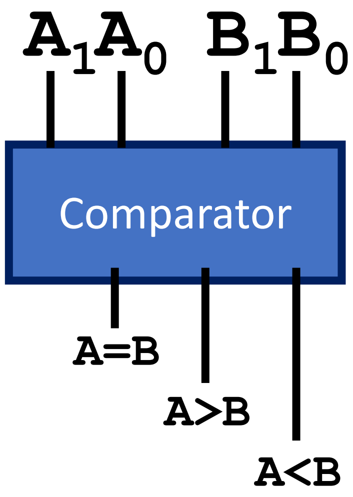
Basic Comparators



- Consider two binary numbers A and B, where A and B are one bit long.
- The circuits for this would be:
  - A==B: $A \cdot B + \bar{A} \cdot \bar{B}$
  - A>B: $A \cdot \bar{B}$
  - A<B: $\bar{A} \cdot B$

A	B
0	0
0	1
1	0
1	1

Basic Comparators



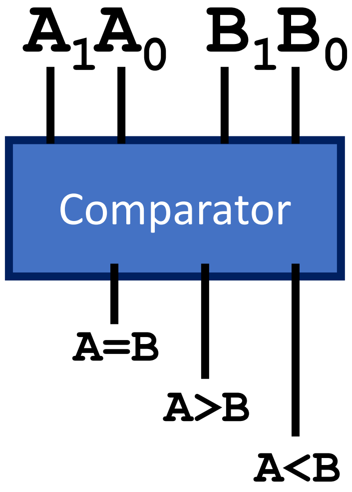
- What if A and B are two bits long?
- The terms for this circuit for have to expand to reflect the second signal.
- For example:
  - A==B:

$$(A_1 \cdot B_1 + \overline{A}_1 \cdot \overline{B}_1) \cdot (A_0 \cdot B_0 + \overline{A}_0 \cdot \overline{B}_0)$$

Make sure that the values of bit 1 are the same

Make sure that the values of bit 0 are the same

Basic Comparators



- What about checking if A is greater than B?

- A>B: $A_1 \cdot \overline{B_1} + (A_1 \cdot B_1 + \overline{A_1} \cdot \overline{B_1}) \cdot (A_0 \cdot \overline{B_0})$

Check if first bit satisfies condition

If not, check that the first bits are equal…

…and then do the 1-bit comparison

- A<B: $\overline{A_1} \cdot B_1 + (A_1 \cdot B_1 + \overline{A_1} \cdot \overline{B_1}) \cdot (\overline{A_0} \cdot B_0)$

A > B if and only if A1 > B1 or (A1 = B1 and A0 > B0)

General Comparators

- The general circuit for comparators requires you to define equations for each case.
- Case #1: Equality
  - If inputs A and B are equal, then all bits must be the same.
  - Define X_i for any digit i:
    - (equality for digit i)

$$X_i = A_i \cdot B_i + \overline{A}_i \cdot \overline{B}_i$$

  - Equality between A and B is defined as:

$$A{=}{=}B \ : \ X_0 \cdot X_1 \cdot \ldots \cdot X_n$$

Comparators

- Case #2: $A > B$
  - The first non-matching bits occur at bit i, where $A_i=1$ and $B_i=0$. All higher bits match.
  - Using the definition for X_i from before:

$$A>B = A_n \cdot \overline{B}_n + X_n \cdot A_{n-1} \cdot \overline{B}_{n-1} + \ldots + A_0 \cdot \overline{B}_0 \cdot \prod_{k=1}^{n} X_k$$

- Case #3: $A < B$
  - The first non-matching bits occur at bit i, where $A_i=0$ and $B_i=1$. Again, all higher bits match.

$$A<B = \overline{A}_n \cdot B_n + X_n \cdot \overline{A}_{n-1} \cdot B_{n-1} + \ldots + \overline{A}_0 \cdot B_0 \cdot \prod_{k=1}^{n} X_k$$

Comparator truth table

- Given two input vectors of size `n=2`, output of circuit is shown at right.

Inputs				Outputs		
A_1	A_0	B_1	B_0	A < B	A = B	A > B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Comparator example (cont'd)

A<B:

	$\overline{B_0}\cdot\overline{B_1}$	$B_0\cdot\overline{B_1}$	$B_0\cdot B_1$	$\overline{B_0}\cdot B_1$
$\overline{A_0}\cdot\overline{A_1}$	0	1	1	1
$A_0\cdot\overline{A_1}$	0	0	1	1
$A_0\cdot A_1$	0	0	0	0
$\overline{A_0}\cdot A_1$	0	0	1	0

$$LT = B_1\cdot\overline{A_1} + B_0\cdot B_1\cdot\overline{A_0} + B_0\cdot\overline{A_0}\cdot\overline{A_1}$$

Comparator example (cont'd)

A=B:

	$\overline{B}_0 \cdot \overline{B}_1$	$B_0 \cdot \overline{B}_1$	$B_0 \cdot B_1$	$\overline{B}_0 \cdot B_1$
$\overline{A}_0 \cdot \overline{A}_1$	1	0	0	0
$A_0 \cdot \overline{A}_1$	0	1	0	0
$A_0 \cdot A_1$	0	0	1	0
$\overline{A}_0 \cdot A_1$	0	0	0	1

$$EQ = \overline{B}_0 \cdot \overline{B}_1 \cdot \overline{A}_0 \cdot \overline{A}_1 + B_0 \cdot \overline{B}_1 \cdot A_0 \cdot \overline{A}_1 + B_0 \cdot B_1 \cdot A_0 \cdot A_1 + \overline{B}_0 \cdot B_1 \cdot \overline{A}_0 \cdot A_1$$

Comparator example (cont'd)

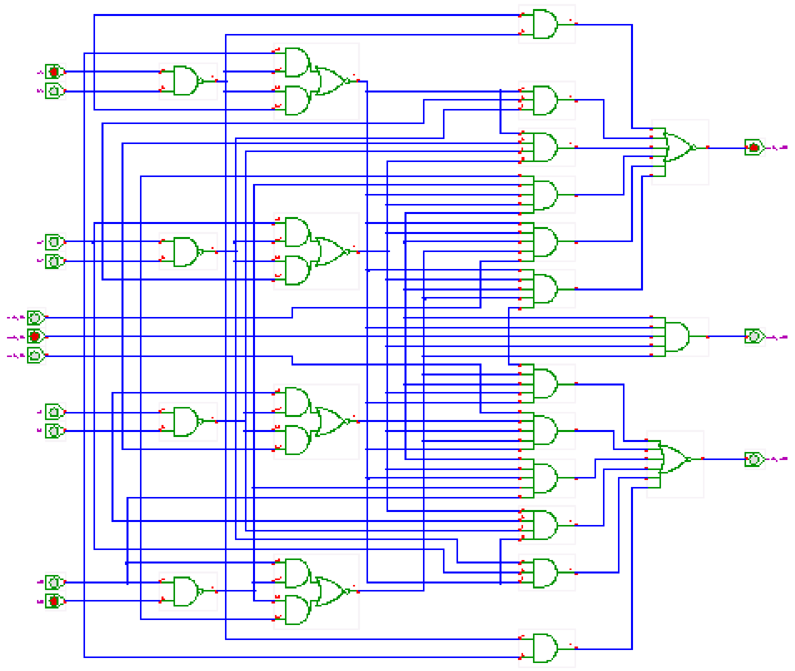
A>B:

	$\overline{B_0} \cdot \overline{B_1}$	$B_0 \cdot \overline{B_1}$	$B_0 \cdot B_1$	$\overline{B_0} \cdot B_1$
$\overline{A_0} \cdot \overline{A_1}$	0	0	0	0
$A_0 \cdot \overline{A_1}$	1	0	0	0
$A_0 \cdot A_1$	1	1	0	1
$\overline{A_0} \cdot A_1$	1	1	0	0

$$GT = \overline{B_1} \cdot A_1 + \overline{B_0} \cdot \overline{B_1} \cdot A_0 + \overline{B_0} \cdot A_0 \cdot A_1$$

Comparing larger numbers

- As numbers get larger, the comparator circuit gets more complex.
- At a certain level, it can be easier sometimes to just process the result of a subtraction operation instead.
  - Easier, less circuitry, just not faster.



Question #1

a) How do you write the number 78 as an 8-bit binary number?

128	64	32	16	8	4	2	1
0	1	0	0	1	1	1	0

b) What is the two's complement of 01101101?

10010011

c) What is the sum of 01101101 and 01101101?

11011010

← Note what's happening here!

Question #2

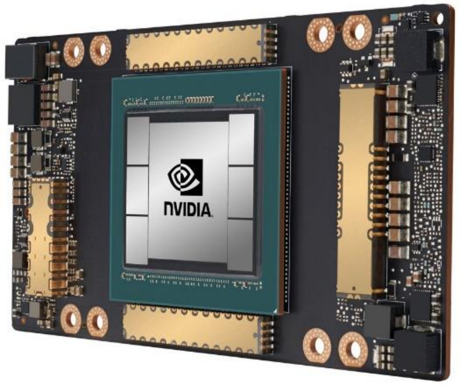
- What groupings are in the K-map on the right?

	$\bar{C}\cdot\bar{D}$	$C\cdot\bar{D}$	$C\cdot D$	$\bar{C}\cdot D$
$\bar{A}\cdot\bar{B}$	1	1	X	1
$A\cdot\bar{B}$	X	0	X	1
$A\cdot B$	1	X	X	1
$\bar{A}\cdot B$	1	X	0	X

- What logic equations do these groupings represent?

$$\bar{A}\cdot\bar{B} + \bar{C}$$

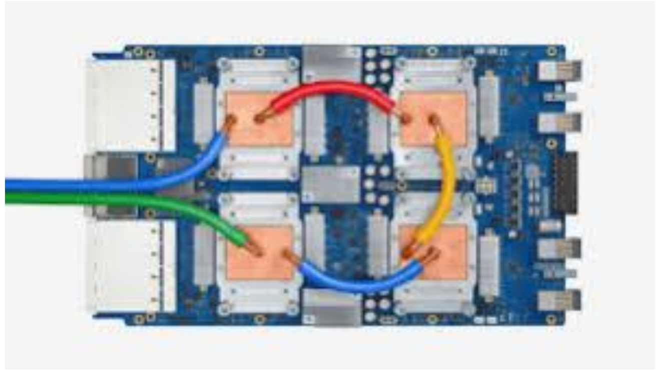
CSCB58:
Computer Organization



Prof. Gennady Pekhimenko

University of Toronto

Fall 2020



The content of this lecture is adapted from the lectures of Larry Zheng and Steve Engels