Digital Comparators

- **Comparator**: A circuit that compares two binary words and indicates whether they are equal
- **Magnitude comparator**: Interprets its inputs as signed or unsigned numbers and indicates their arithmetic relationship (greater or less than)
Example Comparator Use

- Devices are enabled by comparing a “device select” word with a predetermined “device ID”

```
Control Unit
  ▶ binary-coded device select
  ▶ compare device ID
  ▶ device enable
  ▶ ▶ ▶ ▶ ▶
     Device
```

Equality Comparators

- 1-bit comparator
  - Active-high output (DIFF) asserted if the inputs are different

```
A0 1 B0 2
  ▶ ▶
    U1
    DIFF
```

- 4-bit comparator
  - The DIFF output is asserted if any of the input pairs are different

```
A0 1 B0 2
  ▶ ▶
    U1
    DIFF0

A1 4 B1 5
  ▶ ▶
    U1
    DIFF1

A2 9 B2 10
  ▶ ▶
    U1
    DIFF2

A3 12 B3 13
  ▶ ▶
    U1
    DIFF3
```

EQ_L
DIFF
4-Bit Magnitude Comparator

- Two input numbers to compare, 4 bits each: \( A = A_3A_2A_1A_0 \); \( B = B_3B_2B_1B_0 \)
- Three outputs, reporting “greater than”, “less than”, and “equal”, respectively

<table>
<thead>
<tr>
<th>Compared Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_3, B_3 )</td>
<td>( A_2, B_2 )</td>
</tr>
<tr>
<td>( A &gt; B )</td>
<td>( A &lt; B )</td>
</tr>
<tr>
<td>( A &lt; B )</td>
<td>( A &lt; B )</td>
</tr>
<tr>
<td>( A_3 = B_3 )</td>
<td>( A_2 = B_2 )</td>
</tr>
<tr>
<td>( A_3 = B_3 )</td>
<td>( A_2 = B_2 )</td>
</tr>
<tr>
<td>( A &lt; B )</td>
<td>( A &lt; B )</td>
</tr>
<tr>
<td>( A_3 = B_3 )</td>
<td>( A_2 = B_2 )</td>
</tr>
<tr>
<td>( A &lt; B )</td>
<td>( A &lt; B )</td>
</tr>
<tr>
<td>( A_3 = B_3 )</td>
<td>( A_2 = B_2 )</td>
</tr>
</tbody>
</table>

Note “x” (don’t care) notation.

4-Bit Magnitude Comparator

- Input \( A=A_3A_2A_1A_0 \); \( B=B_3B_2B_1B_0 \)
- **Case \( A = B \) :** \( A_3=B_3, A_2=B_2, A_1=B_1, A_0=B_0 \)
  \[ x_i = (A_i \bar{B}_i) \oplus \bar{A}_iB_i \]
  \[ \text{XNOR} = (A_i \bar{B}_i) + (A_i \bar{B}_i) + (A_i \bar{B}_i) \]
  \[ = A_i \bar{B}_i + \bar{A}_iB_i \]
  - Output: \( x_3x_2x_1x_0 \)
- **Case \( A > B \) :**
  - Output: \( A_3 B_3' + x_3 A_2 B_2' + x_3 x_2 A_1 B_1' + x_3 x_2 x_1 A_0 B_0' \)
- **Case \( A < B \) :**
  - Output: \( A_3' B_3 + x_3 A_2' B_2 + x_3 x_2 A_1' B_1' + x_3 x_2 x_1 A_0' B_0' \)
Iterative Combinational Circuits

- General structure: \( n \) identical modules
  - For problems that can be solved by an iterative algorithm:
    1. Set \( C_0 \) to its initial value and set \( i \) to 0
    2. While \( i < n \) repeat:
       a) Use \( C_i \) and \( P_i \) to determine the values of \( P_{O_i} \) and \( C_{i+1} \)
       b) Increment \( i \)

An Iterative Comparator Circuit

- (a) module for one bit
- (b) complete circuit
  - Comparing two \( n \)-bit values \( X \) and \( Y \):
    1. Set \( EQ_0 \) to 1 and set \( i \) to 0
    2. While \( i < n \) repeat:
       a) If \( EQ_i \) is 1 and \( X_i \) equals \( Y_i \), set \( EQ_{i+1} \) to 1
       Else set \( EQ_{i+1} \) to 0
       b) Increment \( i \)
  - Slow because the cascading signals need time to "ripple" from left to right

\[
EQO = (A \oplus B)' \cdot EQI
\]
4-bit Comparator 74x85

- Outputs:
  - Greater-than output (AGTBOUT)
  - Less-than output (ALTBOUT)
  - Equal output (AEQBOUT)

- Cascading inputs:
  - AGTBIN, ALTBIN, AEQBIN

- Cascading inputs and the outputs are arranged in a 1-out-of-3 code, since normally exactly one input and output should be asserted.

10-bit Comparator using 74x85s

AGTBOUT = (A>B) + (A=B) · AGTBIN
AGTBOUT = (A=B) · AEQBIN
AGTBOUT = (A<B) + (A=B) · ALTBIN

XNOR

(A>B) = A3·B3' + (A3·B3)' · A2·B2' + (A3·B3)' · (A2·B2)' · A1·B1' + (A3·B3)' · (A2·B2)' · (A1·B1)' · A0·B0'

![Diagram of 12-bit Comparator using 74x85s](image_url)
### 8-bit Magnitude Comparator

- **74x682**
  - Does not have cascading inputs (unlike 74x85)
  - Does not provide a “less than” output

**Diagram**

- Compares equality using 4 XNOR gates
- Compares if \( P[7–0] > Q[7–0] \)

### Arithmetic Conditions from 74x682

- Not-provided conditions can be implemented as a function of outputs PEQQ_L and PGTQ_L

**Diagram**

- Compares inequality (\( \neq \))
- Compares greater than or equal to (\( \geq \))
- Compares less than or equal to (\( \leq \))

11 of 12

12 of 12