Lecture 4b

Combination Logic & Hardware Issues
Lecture Overview

- Inside logic gates
- TTL and CMOS logic
- Transmission gates
- Tri-state buffer
- Electrical characteristics: Signal levels, Noise margins, Fan-out, Speed
- Reading assignment: Chapter 10.1, 10.2,
Dear ELEC2200 students,

Even though you have 24-hour access to room 3115, please DON'T USE the computers inside room 3115 during the following timeslots (which are being used by ELEC3300 and ELEC4320), EVEN if there are "available computers without any user". The reason is that we cannot disrupt other course labs.

These are the time-slots that you MUST NOT enter into room 3115:
- ELEC3300 (on Tue 6:00pm-7:50pm, Wed 1:00pm-2:50pm) & ELEC4320 (on Mon 12:00pm-5:50pm).

Also, please DON'T USE the computers inside room 3115 during ELEC2200 Lab hours if it’s NOT your lab section.

Yours,

Raymundo Tang Tang

ELEC2200
Inside an IC package
Inside 7401 TTL

7401 TTL integrated circuit (IC) pictured under electron microscope
CMOS transistor

- Transistor as a switch

- Two types of CMOS transistors
  - PMOS and NMOS transistors complement of each other
  - 3-terminals transistor: Gate (G), Source (S) and Drain (D)

- NMOS transistor
  - Circuit Symbol

- PMOS transistor
  - Circuit Symbol
NMOS Transistor (Switch) is ON when input G is HIGH

PMOS Transistor (Switch) is ON when input G is LOW
The CMOS Inverter

- When \( IN = 1 \), NMOS switch is ON (PMOS OFF) \( \rightarrow \) output is pulled DOWN to ground (Low) level (\( Out = 0 \))
- When \( IN = 0 \), PMOS switch is ON (NMOS OFF) \( \rightarrow \) output is pulled UP to Supply (VDD: High) level (\( Out = 0 \))
The CMOS Inverter

- Always connect PMOS (source input) to VDD
- Always connect NMOS (source input) to Ground
- You cannot swap them. Otherwise, the circuit won’t work!
### NAND Gate

**Pull-Up Network (PUN):**
PMOS switches in parallel

**Pull-Down Network (PDN):**
NMOS switches in series

#### Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>PDN</th>
<th>PUN</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Off</td>
<td>On</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Off</td>
<td>On</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Off</td>
<td>On</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>On</td>
<td>Off</td>
<td>0</td>
</tr>
</tbody>
</table>

### NOR Gate

**Pull-Up Network (PUN):**
PMOS switches in series

**Pull-Down Network (PDN):**
NMOS switches in parallel

#### Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>PDN</th>
<th>PUN</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Off</td>
<td>On</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>On</td>
<td>Off</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>On</td>
<td>Off</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>On</td>
<td>Off</td>
<td>0</td>
</tr>
</tbody>
</table>
CMOS Gates for $n$ inputs

- Make sure PUN and PDN are complementary so that they cannot be ON at the same time
- If PMOS switches in series $\rightarrow$ NMOS switches should be in parallel
- If PMOS switches in parallel $\rightarrow$ NMOS switches should be in series

Pull-Up Network (PUN): PMOS switches

Pull-Down Network (PDN): NMOS switches

Turn on PUN switches if you want the output to be logic 1

Turn on PDN switches if you want the output to be logic 0
CMOS Basic Building Blocks

- NOT, NAND and NOR gates
  - Notice the outputs are inverted in all CMOS building blocks
  - A 2 input NAND/NOR gate (equivalent gate) requires 4 transistors (switches)
CMOS Realization of NOT (INVERTER)

Implement

\[ f = \overline{x} \]

Note input to PMOS is complemented

Implement \[ \overline{f} = x \]
CMOS Realization of NOR

AND operation requires transistors in series
OR operation requires transistors in parallel

Implement

\[ f = x_1 + x_2 = \bar{x}_1 \bar{x}_2 \]

Note input to PMOS is complemented
PMOS are in series (x1’ AND x2’)

Implement \( \bar{f} = x_1 + x_2 \)
NMOS are in series (x1 OR x2)
CMOS Realization of NAND

AND operation requires transistors in series
OR operation requires transistors in parallel

Implement \( f = \overline{x_1x_2} \)

Note input to PMOS is complemented. PMOS are in parallel (\( x_1' \) OR \( x_2' \))

Implement \( \overline{f} = x_1x_2 \)

NMOS are in series (\( x_1 \) AND \( x_2 \))
Example 1

Implement the below function using CMOS logic

\[ f = \overline{x}_1 + \overline{x}_2 \overline{x}_3 \]
Example 1

Implement \( f = \overline{x_1} + \overline{x_2}\overline{x_3} \)

All the inputs are already in complement \( x_1' \) parallel with \( (x_2' \text{ and } x_3' \text{ in series}) \)

Implement \( \overline{f} = \overline{x_1} + \overline{x_2}\overline{x_3} = x_1(x_2 + x_3) \)

\( x_1 \) in series with \( (x_2 \text{ and } x_3 \text{ in parallel}) \)
Example 2

Implement the below function using CMOS logic

\[ f = \overline{x}_1 + (\overline{x}_2 + \overline{x}_3) \overline{x}_4 \]
Example 2

Implement

\[ f = \bar{x}_1 + (\bar{x}_2 + \bar{x}_3)x_4 \]

All the inputs are already in complement \( x_1' \) in parallel with \([(x_2' \text{ and } x_3') \text{ in parallel}] in series with \( x_4' \)]

Implement

\[ \bar{f} = x_1(x_2x_3 + x_4) \]

\( x_1 \) in series with \([(x_2 \text{ and } x_3 \text{ in series}) \text{ in parallel with } x_4] \)
Another way: Pass Transistor Logic

- Use PMOS or NMOS transistors to pass bits (1 or 0) from input to output
- Control the gate (G) input to pass or not to pass the bit
- Either Source (S) or Drain (D) can be used for input or output (bi-directional)
- Notice that there is no supply (V_{DD}) or GND connection at all
- NMOS works well connecting “logic 0”
- PMOS works well connecting “logic 1”

Not Efficient!
Transmission Gate (is also a switch)

- To pass both 0 and 1 efficiently
  - Connect one NMOS and one PMOS like this →
  - Label G (gate) inputs as complementary Select (S) inputs
    - NMOS switches work well only when connecting to 0
    - PMOS switches work well only when connecting to 1

Other symbols to show a Transmission Gate
Multiplexer by CMOS Transmission gate (TG)

- **2-to-1 multiplexer**

  ![Diagram of a 2-to-1 multiplexer using CMOS transmission gates](image)

  2 TG and 1 inverter
  - 1.5 equivalent gates

  "Combinational multiplexer"
  Implemented with NAND gates
  - 3.5 equivalent gates
Multiplexer by CMOS Transmission gate (TG)

- **4-1 multiplexer**
  - (2+4) TG and 2 inverters
    - 4 equivalent gates
  - by NAND-NAND gate
    - 9 equivalent gates

3 input NAND/NOR gate = 1.5 Equivalent gate
4 input NAND/NOR gate = 2 Equivalent gate
Each inverter = 0.5 Equivalent gate
Total = \((4 \times 1.5) + (1 \times 2) + (2 \times 0.5) = 9\)
Demultiplexer by Transmission gates

- 1-2 demultiplexer
  - 1 select input
  - 1 enable input
  - 2 outputs
    » Y goes to 1 output
    » the other is floated

![Diagram of a 1-2 demultiplexer with transmission gates.]
Demultiplexer by Transmission gates

- **1-2 demultiplexer**
  - 1 select input
  - 1 enable input $Y$
    - 1 logic 0 input
  - 2 outputs
    - $Y$ goes to 1 output
    - 0 goes to the other
  - 4 TGs and 1 NOT
    - 2.5 equivalent gates

![Diagram of 1-2 demultiplexer with transmission gates and logic symbols]
**Tri-state gate = Buffered TG**

- Using Transmission gates (TGs) is good because it cuts down number of transistors.
- But, signal becomes weak when TGs are cascaded! TGs do not have supply (VDD)!
- Tri-state gate solves the problem with a buffered TG (Buffer has the supply VDD)!

![Diagram of tri-state gate]

- When EN=1, then Y=A
- When EN=0, then Y is disconnected from A and Y is *undriven*, we call it the high-impedance state (State ‘Z’), so Y = ‘Z’, at high impedance state.
- Tri-state means now Y can be at 3 different states: 0, 1, ‘Z’
Tri-state Gates

**Inverter**

**Tri-state Inverters**

**Tri-state Buffer**

---

### Inverter

![Inverter Diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Tri-state Inverters

![Tri-state Inverters Diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>OE</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>Z</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Tri-state Buffer

![Tri-state Buffer Diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>EN</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>Z</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Using Tri-state gates

- The output of tri-state gates can be tied together.
  - As long as only one output has value (0 or 1) and the other outputs are at high impedance states ('Z')
- Must make sure that ENA and ENB are not both ‘1’ at the same time, or Y will be driven from multiple active sources

<table>
<thead>
<tr>
<th>ENB</th>
<th>ENA</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Z (high impedance)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>A</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X (very bad idea!!!!)</td>
</tr>
</tbody>
</table>
Multiplexer by Tri-state Gates

• 2-to-1 multiplexer
  – 2 tri-state gates
    » only one is enabled

• 4-to-1 multiplexer
In digital circuits, digitally-encoded information is represented by means of two distinct voltage ranges:

- **Logic 0:** \( V_{\text{MIN}} \leq V \leq V_{\text{OL}} \)

- **Logic 1:** \( V_{\text{OH}} \leq V \leq V_{\text{MAX}} \)

- **Undefined logic value:** \( V_{\text{OL}} \leq V \leq V_{\text{OH}} \)
Define **switching point** or **logic threshold**:

- \( V_M \equiv \text{input voltage for which } V_{\text{OUT}} = V_{\text{IN}} \)
  - For \( 0 \leq V_{\text{IN}} < V_M \) \( \Rightarrow V_{\text{OUT}} = V^+ \)
  - For \( V_M < V_{\text{IN}} \leq V^+ \) \( \Rightarrow V_{\text{OUT}} = 0 \)
Input-Output relation (Real Inverter)

- **Logic 0:**
  - $V_{\text{MIN}} \equiv$ output voltage for which $V_{\text{IN}} = V^+$
  - $V_{\text{OL}} \equiv$ smallest output voltage where slope $= -1$

- **Logic 1:**
  - $V_{\text{OH}} \equiv$ largest output voltage where slope $= -1$
  - $V_{\text{MAX}} \equiv$ output voltage for which $V_{\text{IN}} = 0$
Two other important voltages:

\[ V_{IL} = \text{smallest input voltage where slope } = -1 \]
\[ V_{IH} = \text{highest input voltage where slope } = -1 \]
Noise Margin (NM)

Noise is undesired and causes output voltage (1) to change its value (diagram). As long as the new value (2) is within the noise margin, result will be correct!

\[
\text{NM}_{\text{HIGH}} = V_{\text{OH}} - V_{\text{IH}} \\
\text{NM}_{\text{LOW}} = V_{\text{IL}} - V_{\text{OL}}
\]

Fig. 10-5 Signals for Evaluating Noise Margin
Circuit Characteristics – Propagation Delay

- Propagation delay: time for signal from input to output
  - $t_{PHL}$: for output to be changed from High to Low
  - $t_{PLH}$: for output to be changed from Low to High
Propagation Delay

- Cascaded gates:

In general:
prop. delay = sum of individual prop. delays of gates in series.
Hazards/Glitches

- Hazards/glitches: unwanted switching at the outputs
  - Occur when different paths through circuit have different propagation delays
    - As in pulse shaping circuits we just analyzed
  - Dangerous if logic causes an action while output is unstable
    - May need to guarantee absence of glitches
- Usual solutions
  1) Wait until signals are stable (by using a clock): preferable (easiest to design when there is a clock – synchronous design)
  2) Design hazard-free circuits: sometimes necessary (clock not used – asynchronous design)
Types of Hazards

- Static 1-hazard
  - Input change causes output to go from 1 to 0 to 1

- Static 0-hazard
  - INput change causes output to go from 0 to 1 to 0

- Dynamic hazards
  - Input change causes a double change from 0 to 1 to 0 to 1 OR from 1 to 0 to 1 to 0
Static Hazards

- Due to a literal and its complement momentarily taking on the same value
  - Thru different paths with different delays and reconverging
  - May cause an output that should have stayed at the same value to momentarily take on the wrong value

Example:

When \( S \) changes from 0 to 1, \( F \) becomes 1 for a moment.

\[ F = SS' = 0 \]

New value \( S' \to 0 \) has not reached there yet because the inverter delay!
Dynamic Hazards

- Due to the same versions of a literal taking on opposite values
  - Thru different paths with different delays and reconverging
- May cause an output that was to change value to change 3 times instead of once
- Example: When B changes from 0 to 1

```
When B changes from 0 to 1
```

```
A  0  0 -> 1  1 -> 0  0 -> 1  1 -> 0  1 -> 0  1 -> 0  1 -> 0
B  0  0 -> 1  0 -> 1  0 -> 1  0 -> 1  0 -> 1  0 -> 1
C  1  0  0  0  0  0  0  0

dynamic hazards
```
**Circuit Characteristics - Fan-out**

- How many gates can be driven by one gate...
- \( I_{OH} = 400 \, \mu A, I_{IH} = 40 \, \mu A, \text{Fan-out} = \frac{I_{OH}}{I_{IH}} = 10 \)
- \( I_{OL} = 16 \, mA, I_{IL} = 1.6 \, mA, \text{Fan-out} = \frac{I_{OL}}{I_{IL}} = 10 \)
TTL logic gates....

- Built using Bipolar Transistors, and using 5V supply (Vcc)
- The circuit is a bit more complicated, don’t worry if you do not understand the circuit. You will be taught again later.
  - Here shows an example of a 3-input NAND gate
Common TTL Logic Gates

Fig. 11-1 Digital Gates in IC Packages with Identification Numbers and Pin Assignments
Common TTL Logic Gates

- 3-input NAND: 7410
- 4-input NAND: 7420
- 2-input OR: 7432
- 2-input XOR: 7486

Fig. 11-1: Digital Gates in IC Packages with Identification Numbers and Pin Assignments
## TTL Functional Selection

### Functional Selection

#### Abbreviations

- **S**: Synchronous
- **A**: Asynchronous
- **B**: Both Synchronous and Asynchronous
- **2S**: 2-State Output
- **3S**: 3-State Output
- **OC**: Open-Collector Output
- **P**: Planned (See FAST/LS Selector Guide, SG-60 for latest availability status)
- **X**: Available

#### Inverters

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hex</td>
<td>2S</td>
<td>04</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>OC</td>
<td>05</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

#### AND Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>2S</td>
<td>08</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>OC</td>
<td>09</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Triple 2-Input</td>
<td>2S</td>
<td>11</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>OC</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dual 4-Input</td>
<td>2S</td>
<td>21</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

#### NAND Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2S</td>
<td>51</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

#### Exclusive OR Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>2S</td>
<td>86</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>OC</td>
<td>136</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2S</td>
<td>386</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Exclusive NOR Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>OC</td>
<td>266</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

#### AND-OR-INVERT Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual 2-Wide, 2-Input 3-Input</td>
<td>2S</td>
<td>51</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
# TTL Functional Selection

## NAND Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>2S</td>
<td>00</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OC</td>
<td>01</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>03</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quad 2-Input, High Voltage</td>
<td>2S</td>
<td>26</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>10</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>12</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triple 3-Input</td>
<td>2S</td>
<td>20</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>22</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dual 4-Input</td>
<td>2S</td>
<td>30</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OC</td>
<td>32</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8-Input</td>
<td>2S</td>
<td>133</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

## OR Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>2S</td>
<td>32</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

## NOR Gates

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quad 2-Input</td>
<td>2S</td>
<td>02</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Triple 3-Input</td>
<td>2S</td>
<td>27</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual 5-Input</td>
<td>2S</td>
<td>200</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

## Schmitt Triggers

<table>
<thead>
<tr>
<th>Description</th>
<th>Type of Output</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual 2-Wide, 2-Input 3-Input</td>
<td>2S</td>
<td>51</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4-Wide, 2-3-2-3-Input</td>
<td>2S</td>
<td>54</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2-Wide, 4-Input</td>
<td>2S</td>
<td>55</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4-Wide, 4-2-2-3-Input</td>
<td>2S</td>
<td>64</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

## SSI Flip-Flops

<table>
<thead>
<tr>
<th>Description</th>
<th>Clock Edge</th>
<th>No.</th>
<th>LS</th>
<th>FAST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dual D w/Set &amp; Clear</td>
<td>Pos</td>
<td>74</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual D w/Set &amp; Clear</td>
<td>Pos</td>
<td>74A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Set</td>
<td>Neg</td>
<td>113A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Reset</td>
<td>Neg</td>
<td>73A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Same as 73A with Different Pinout</td>
<td>Neg</td>
<td>107A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Set &amp; Clear Individual J, K, CP, SP, CD Inputs</td>
<td>Neg</td>
<td>76A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Same as 76 with Different Pinout</td>
<td>Neg</td>
<td>112</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Reset</td>
<td>Neg</td>
<td>112A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Reset</td>
<td>Neg</td>
<td>114A</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Set &amp; Clear</td>
<td>Fos</td>
<td>109</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Dual JK w/Reset</td>
<td>Fos</td>
<td>109A</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Example implementation by the TTL ICs

• How many TTL ICs are required to implement

  1. one 3-input OR gate?
     - One; use 2 2-input OR gates in one 74LS32
     - Two; use one 3-input NOR gate (74LS27) followed by an inverter (74LS04)

  2. One 2-input AND gate and one 3-input AND gate?
     - One; use 3 2-input AND gates (74LS08) or 2 3-input AND gates (74LS11)
     - Two; use one 2-input AND gate (74LS08) and one 3-input AND gate (74LS11)

  3. One inverter, one 2-input NAND gate and one 3-input NAND gate?
     - Three; use one inverter (74LS04), one 2-input NAND gate (74LS00) and one 3-input NAND gate (74LS10)
     - Two;
     - One?
Reading assignment

- Chapter 10.1 to 10.2 of textbook