

**DIGITAL SYSTEM DESIGN APPLICATION**

**EHB436E CRN: 11280**

**Salih Ömer Ongün**

**040220780**

**Experiment 2**

**DECODER**

**Decoder Truth Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I3** | **I2** | **I1** | **I0** | **O0** | **O1** | **O2** | **O3** | **O4** | **O5** | **O6** | **O7** | **O8** | **O9** | **O10** | **O11** | **O12** | **O13** | **O14** | **O15** |
| **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| **1** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** |
| **1** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |

**Design Sources**

MSI\_Library file design source

**module** DECODER

**(**

**input** **[**3**:**0**]** IN**,**

**output** **[**15**:**0**]** OUT

**);**

**reg** **[**15**:**0**]** out\_reg**;**

**always@(\*)**

**begin**

**case(**IN**)**

4'b0000**:** out\_reg **=** 16'b0000\_0000\_0000\_0001**;**

4'b0001**:** out\_reg **=** 16'b0000\_0000\_0000\_0010**;**

4'b0010**:** out\_reg **=** 16'b0000\_0000\_0000\_0100**;**

4'b0011**:** out\_reg **=** 16'b0000\_0000\_0000\_1000**;**

4'b0100**:** out\_reg **=** 16'b0000\_0000\_0001\_0000**;**

4'b0101**:** out\_reg **=** 16'b0000\_0000\_0010\_0000**;**

4'b0110**:** out\_reg **=** 16'b0000\_0000\_0100\_0000**;**

4'b0111**:** out\_reg **=** 16'b0000\_0000\_1000\_0000**;**

4'b1000**:** out\_reg **=** 16'b0000\_0001\_0000\_0000**;**

4'b1001**:** out\_reg **=** 16'b0000\_0010\_0000\_0000**;**

4'b1010**:** out\_reg **=** 16'b0000\_0100\_0000\_0000**;**

4'b1011**:** out\_reg **=** 16'b0000\_1000\_0000\_0000**;**

4'b1100**:** out\_reg **=** 16'b0001\_0000\_0000\_0000**;**

4'b1101**:** out\_reg **=** 16'b0010\_0000\_0000\_0000**;**

4'b1110**:** out\_reg **=** 16'b0100\_0000\_0000\_0000**;**

4'b1111**:** out\_reg **=** 16'b1000\_0000\_0000\_0000**;**

**default:** out\_reg **=** 16'b0000\_0000\_0000\_0000**;**

**endcase**

**end**

**assign** OUT **=** out\_reg**;**

**endmodule**

**module** top\_module

**(**

**input** **[**7**:**0**]** sw**,**

**input** **[**3**:**0**]** btn**,**

**output** **[**7**:**0**]** led**,**

**output** **[**6**:**0**]** cat**,**

**output** **[**3**:**0**]** an**,**

**output** **[**0**:**0**]** dp

**);**

DECODER decoder1

**(**

**.**IN**(**sw**[**3**:**0**]),**

**.**OUT**({**dp**,** cat**,** led**})**

**);**

**assign** an **=** 4'b1110**;**

top\_module file design source

**Simulation Sources**

top\_module file simulation source

**module** top\_module\_tb**();**

**reg** **[**7**:**0**]** SW**=** 8'b0**;**

**reg** **[**3**:**0**]** BTN**=** 4'b0**;**

**wire** **[**7**:**0**]** LED**;**

**wire** **[**6**:**0**]** CAT**;**

**wire** **[**3**:**0**]** AN**;**

**wire** **[**0**:**0**]** DP**;**

**integer** i**;**

top\_module uut

**(**

**.**sw**(**SW**),**

**.**btn**(**BTN**),**

**.**led**(**LED**),**

**.**cat**(**CAT**),**

**.**an**(**AN**),**

**.**dp**(**DP**)**

**);**

**initial**

**begin**

SW**[**3**:**0**]** **=** 4'b0000**;** // decoder ve encoder

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0001**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0010**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0011**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0100**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0101**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b0110**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b0111**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1000**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1001**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1010**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1011**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1100**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1101**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1110**;**

**#**10**;**

SW**[**3**:**0**]** **=**4'b1111**;**

**#**10**;**

**Simulation Wave**

Behavioral simulation wave screenshot

ekran görüntüsü, multimedya yazılımı, grafik yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturulduekran görüntüsü, çizgi, diyagram, renklilik içeren bir resim

Açıklama otomatik olarak oluşturuldu

As can be seen from the simulation wave, the design works correctly. In the "Decoder" truth table, the LED, CAT and AN outputs are given values, respectively, as they should be

**RTL Schematic**

metin, diyagram, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Technology Schematic**

metin, diyagram, çizgi, paralel içeren bir resim

Açıklama otomatik olarak oluşturuldu

diyagram, plan, çizgi, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

There are 16 LUTs in decoder. There are 7 LUTs for "cat", 1 for "dp" and 8 for "led". These LUTs are connected to "OUT". "dp" has the most significant bit, "led" has the least significant bit. One bit of "OUT" has a logical value of 1 according to the input statements. These LUTs have values ​​according to their "OUT" statements.

**Timing Report**

**metin, ekran görüntüsü, menü, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, sayı, numara, menü içeren bir resim

Açıklama otomatik olarak oluşturuldu**

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

The greatest delay of DECODER is 10.562ns.

**PRIORITY ENCODER**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **IN[3]** | **IN[2]** | **IN[1]** | **IN[0]** | **O[1]** | **O[0]** | **V** |
| **0** | **0** | **0** | **0** | **X** | **X** | **0** |
| **0** | **0** | **0** | **1** | **0** | **0** | **1** |
| **0** | **0** | **1** | **X** | **0** | **1** | **1** |
| **0** | **1** | **X** | **X** | **1** | **0** | **1** |
| **1** | **X** | **X** | **X** | **1** | **1** | **1** |

**Karnaugh Map Hand Drawing**

**metin, çizim, taslak, el yazısı içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, taslak, çizim, çizgi sanatı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Design Sources**

MSI\_Library file design source

**module** ENCODER

**(**

**input** **[**3**:**0**]** IN**,**

**output** **[**1**:**0**]** OUT**,**

**output** V

**);**

**assign** V **=** IN**[**0**]** **|** IN**[**1**]** **|** IN**[**2**]** **|** IN**[**3**]** **;**

**assign** OUT**[**0**]** **=** IN**[**3**]** **|** **(** IN**[**1**]** **&** **~(**IN**[**2**])** **)** **;**

**assign** OUT**[**1**]** **=** IN**[**3**]** **|** IN**[**2**]** **;**

**endmodule**

top\_module file design source

ENCODER encoder1

**(**

**.**IN**(**sw**[**3**:**0**]),**

**.**OUT**(**led**[**1**:**0**]),**

**.**V**(**led**[**7**])**

**);**

**Simulation Sources**

Encoder simulation source is same as decoder simulation source.

**Simulation Wave**

Behavioral simulation wave screenshot

**ekran görüntüsü, multimedya yazılımı, grafik yazılımı, düzeltme, tashih içeren bir resim

Açıklama otomatik olarak oluşturuldu**

As can be seen from the simulation wave, the design works correctly. The V output (LED[7]) gives the value 1 for all inputs except the 4-bit 0 input. The OUT[0] and OUT[1] outputs, which are connected to LED[0] and LED[1] respectively, are working correctly as the truth table should. Since nothing is connected to the other outputs, they give the Z value.

**RTL Schematic**

**metin, diyagram, çizgi, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Technology Schematic**

metin, diyagram, plan, şematik içeren bir resim

Açıklama otomatik olarak oluşturuldu

RTL schematics represent the project with gates. There are 6 gates in this implementation. However, Technology Schematic represents the project with Fpga sources. There are 3 LUTs in this implementation. There are not any gates in Fpga. FPGAs are implemented with applications such as LUTs.

**Timing Report**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

The greatest delay of ENCODER is 9.674ns.

**Utilization Report**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Always- Case Design Code**

**module** ENCODER

**(**

**input** **[**3**:**0**]** IN**,**

**output** **[**1**:**0**]** OUT**,**

**output** V

**);**

**reg** V\_reg**;**

**reg** **[**1**:**0**]** OUT\_reg**;**

**always@(\*)**

**begin**

**casex(**IN**)**

4'b0000**:**

**begin**

V\_reg **=** 0**;**

OUT\_reg **=** 2'bxx**;**

**end**

4'b0001**:**

**begin**

V\_reg **=** 1**;**

OUT\_reg **=** 2'b00**;**

**end**

4'b001x**:**

**begin**

V\_reg **=** 1**;**

OUT\_reg **=** 2'b01**;**

**end**

4'b01xx**:**

**begin**

V\_reg **=** 1**;**

OUT\_reg **=** 2'b10**;**

**end**

4'b1xxx**:**

**begin**

V\_reg **=** 1**;**

OUT\_reg **=** 2'b11**;**

**end**

**endcase**

**end**

**assign** OUT **=** OUT\_reg**;**

**assign** V **=** V\_reg**;**

**endmodule**

**Timing Report**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Utilization Report**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Two ENCODER designs have the same source consumption. However, their delay durations are not same. Behavioral design has less delay for "V" out, structural design have less delay for "OUT[0]" output. Output "OUT[1]" has the same delay for both designs.

**MULTIPLEXER**

**Design Sources**

MSI\_Library file design source

**module** MUX

**(**

**input** **[**3**:**0**]** D**,**

**input** **[**1**:**0**]** S**,**

**output** O

**);**

**assign** O **=** **(**S**[**0**]** **==** 1'b0**)** **?**

**(**S**[**1**]** **==** 1'b0**)** **?** D**[**0**]** **:** D**[**2**]:**

**(**S**[**1**]** **==** 1'b0**)** **?** D**[**1**]** **:** D**[**3**];**

**endmodule**

top\_module file design source

MUX mux1

**(**

**.**D**(**sw**[**3**:**0**]),**

**.**S**(**btn**[**1**:**0**]),**

**.**O**(**led**[**0**])**

**);**

**Simulation Sources**

top\_module file simulation source

SW**[**3**:**0**]** **=** 4'b0001**;** // mux

BTN**[**1**:**0**]** **=** 2'b00**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b1101**;**

BTN**[**1**:**0**]** **=** 2'b01**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0100**;**

BTN**[**1**:**0**]** **=** 2'b010**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0111**;**

BTN**[**1**:**0**]** **=** 2'b11**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b1110**;**

BTN**[**1**:**0**]** **=** 2'b00**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b0010**;**

BTN**[**1**:**0**]** **=** 2'b01**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b1011**;**

BTN**[**1**:**0**]** **=** 2'b010**;**

**#**10**;**

SW**[**3**:**0**]** **=** 4'b1000**;**

BTN**[**1**:**0**]** **=** 2'b11**;**

**#**10**;**

**Simulation Wave**

Behavioral simulation wave screenshot

ekran görüntüsü, multimedya yazılımı, renklilik, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

As can be seen from the simulation wave, the design works correctly. The pink glowing part is our output value connected to LED[0]. Our control inputs are connected to the yellow BTN values. When we check the BTN values ​​according to the truth table, the output gives the value entered at the input.

**RTL Schematic**

**metin, diyagram, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Technology Schematic**

**metin, diyagram, plan, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Timing Report**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Utilization Report**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**Always- Case Design Code**

**module** MUX

**(**

**input** **[**3**:**0**]** D**,**

**input** **[**1**:**0**]** S**,**

**output** O

**);**

**reg** O\_reg**;**

**always@(\*)**

**begin**

**case(**S**)**

2'b00 **:** O\_reg **=** D**[**0**];**

2'b01 **:** O\_reg **=** D**[**1**];**

2'b10 **:** O\_reg **=** D**[**2**];**

2'b11 **:** O\_reg **=** D**[**3**];**

**endcase**

**end**

**assign** O **=** O\_reg**;**

**endmodule**

**Timing Report**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Utilization Report**

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Two MUX designs have the same source consumption. However, there are big differences between the two implementations in terms of the delay of each path. Obviously, always and **case** blocks have less delay than assign block.

**DEMULTIPLEXER**

**Design Sources**

MSI\_Library file design source

**module** DEMUX

**(**

**input** D**,**

**input** **[**1**:**0**]** S**,**

**output** **[**3**:**0**]** O

**);**

**wire** and\_o\_0**,**and\_o\_1**,**and\_o\_2**,**and\_o\_3**;**

**wire** s0\_not\_o**,**s1\_not\_o**;**

NOT s0\_not

**(**

**.**I**(**S**[**0**]),**

**.**O**(**s0\_not\_o**)**

**);**

NOT s1\_not

**(**

**.**I**(**S**[**1**]),**

**.**O**(**s1\_not\_o**)**

**);**

AND and0

**(**

**.**l1**(**s0\_not\_o**),**

**.**l2**(**s1\_not\_o**),**

**.**O**(**and\_o\_0**)**

**);**

AND and1

**(**

**.**l1**(**S**[**0**]),**

**.**l2**(**s1\_not\_o**),**

**.**O**(**and\_o\_1**)**

**);**

AND and2

**(**

**.**l1**(**s0\_not\_o**),**

**.**l2**(**S**[**1**]),**

**.**O**(**and\_o\_2**)**

**);**

AND and3

**(**

**.**l1**(**S**[**0**]),**

**.**l2**(**S**[**1**]),**

**.**O**(**and\_o\_3**)**

**);**

TRI tri\_0

**(**

**.**I**(**D**),**

**.**E**(**and\_o\_0**),**

**.**O**(**O**[**0**])**

**);**

TRI tri\_1

**(**

**.**I**(**D**),**

**.**E**(**and\_o\_1**),**

**.**O**(**O**[**1**])**

**);**

TRI tri\_2

**(**

**.**I**(**D**),**

**.**E**(**and\_o\_2**),**

**.**O**(**O**[**2**])**

**);**

TRI tri\_3

**(**

**.**I**(**D**),**

**.**E**(**and\_o\_3**),**

**.**O**(**O**[**3**])**

**);**

**endmodule**

top\_module file design source

DEMUX demux1

**(**

**.**D**(**sw**[**0**]),**

**.**S**(**btn**[**1**:**0**]),**

**.**O**(**led**[**3**:**0**])**

**);**

**Simulation Sources**

top\_module file simulation source

SW**[**0**]** **=** 1**;** // demux

BTN**[**1**:**0**]** **=** 2'b00**;**

**#**10**;**

SW**[**0**]** **=** 1**;**

BTN**[**1**:**0**]** **=** 2'b01**;**

**#**10**;**

SW**[**0**]** **=** 1**;**

BTN**[**1**:**0**]** **=** 2'b10**;**

**#**10**;**

SW**[**0**]** **=** 1**;**

BTN**[**1**:**0**]** **=** 2'b11**;**

**#**10**;**

$finish**;**

**Simulation Wave**

Behavioral simulation wave screenshot

ekran görüntüsü, renklilik, multimedya yazılımı, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

As can be seen from the simulation wave, the design works correctly. The value 1 at the top of the image is the input, which is SW[0]. The values ​​in yellow are the BTNs which are control inputs. BTNs, which are given values ​​according to the truth table values, turn on an LED, which is an output, at each step as it should be. It gives the input value to the LED. Since no value is given to other LEDs, it takes the Z value. The lit LEDs are pink.

**RTL Schematic**

metin, diyagram, plan, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Technology Schematic**

metin, diyagram, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**RESEARCH**

1. Essentially, latches hold data like flip-flops. However, there are major differences between latches and flip-flops. Flip-flops are edge-triggered components on contrast latches are not edge-triggered. Flip-flops are triggered by the clock signal. Therefore, many flip-flops are controlled simultaneously by the same clock signal. There is an example of D latch in below.

metin, ekran görüntüsü, yazılım içeren bir resim

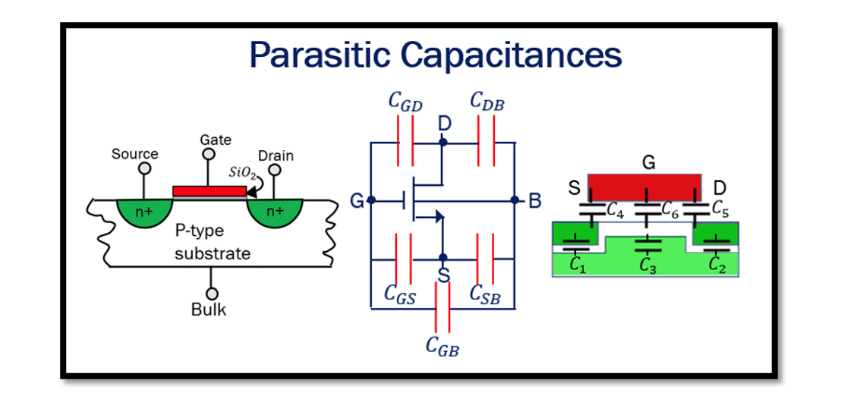
Açıklama otomatik olarak oluşturuldu

D latches are controlled by enable signal. If the enable signal is logical 1, Q is loaded with the value D. If the enable signal is logical 0, Q is unchanged. Therefore, controlling latches is more difficult than controlling flip-flops. Due to this asynchronization, timing issues occur. This timing issue can cause glitches and unnecessary source usage. In addition to these issues, some fpgas may not be able to perform synthesis.

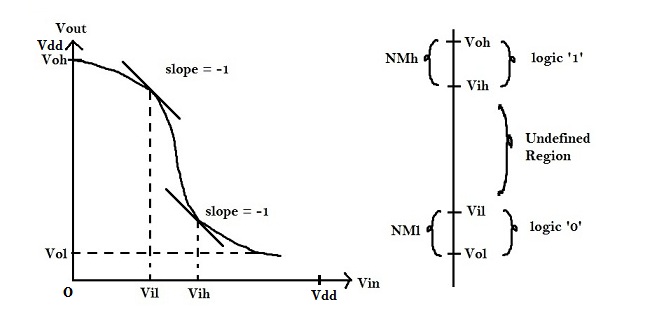
The design should be done with flip-flops. The designer must pay attention to each condition in the if-else and case structure. Assigning else and default condition to if, case structure can resolve the occurrence of the latch.

1. Leading zero is the number of zeros from the most significant bit to the least significant bit unless logic 1 is encountered. The leading zero counter is used for floating-point and advanced microprocessors. The circuits are divided into 4-bit priority encoder circuits. The output of each priority encoder is combined. Using priority encoder has advantages for leading zero counter. After most significant logic 1, other bits are do not care. Therefore, there are logic zeros and an logic one for the priority encoder from the most significant bit to the least significant bits. In this way, it is not difficult to calculate logical zeros with the priority encoder.
2. Fan-in refers to the maximum input that can be connected to a logic gate. Manufacturers write the fan-in number in their data sheets. Using more inputs than the fan-in numbers will result in additional power consumption and delays.

Fan-out refers to the maximum output that can be connected to a logic gate. Manufacturers write the fan-out number in their data sheets. Using more output than the fan-out numbers will result in additional power consumption and output swings.



Logic gates are made up of transistors. Transistors have parasitic capacitances. These capacitances arise from the nature of the semiconductor structure of the transistors. These parasitic capacitances show capacitive effect. If the user connects more outputs than the fan-out number, these parasitic capacitances cause current drops and swings.



There is no exact value for logic 1 output or logic 0 output. There are three ranges for inverters (the building block of digital electronic circuits). If there are outputs more than fan-out number, it will cause current swings. Therefore, the output signal may drop below the logic 1 range.