디지털 논리2 및

컴퓨터 공학 기초 실험2   
보고서

과제제목: Latch & flip-flop design with/without reset/set,

Traffic Light Controller with/without Left Turn Signals

실험일자: 2017년 09월 22일 (금)

제출일자: 2017년 09월 28일 (목)

학 과: 컴퓨터공학과

담당교수: 공진흥 교수님

실습분반: 금요일 5, 6, 7

학 번: 2012722028

성 명: 장 한 별

1. 제목 및 목적
   1. 제목

(1) Latch & flip-flop design with/without reset/set,

(2) Traffic Light Controller with/without Left Turn Signals

* 1. 목적

1. 저장소자 역할을 하는 Latch와 Filp Flop의 동작원리를 이해하고 이를 구현하며 차이점을 확인한다. 또한 set과 reset의 기능을 구현할 수 있도록 한다. 구현한 Flip Flop을 이용하여 register를 설계하고 나아가 synchronous set/reset D Flip Flop과 asynchronous set/reset D Flip Flop의 차이점을 이해할 수 있도록 한다.
2. FSM의 기법 중 하나인 Moore FSM을 이용하여 신호등을 제어하는 logic을 구현, Traffic Light Controller with/without Left Turn Signals를 설계한다. 이때 구현하고자 하는 신호등은 신호등이 시간에 따라 변하는 신호등이 아닌 거리에 차량이 있을 때, 신호등이 Green 이 되고, 거리에 차량이 없을 때는 Red 가 되는 신호등 이다.
3. 원리(배경지식)
4. **Latch & flip-flop design with/without reset/set**
5. SR Latch

|  |
| --- |
|  |
| 위 그림은 SR-Latch 의 symbol이다. SR- latch는 가장 기본적인 저장소자이다. Set 과 Reset의 의미를 포함하고 있다.. |

1. D latch

|  |
| --- |
| C:\Users\Administrator\Desktop\123.jpg |
| 위 그림은 D-Latch 의 symbol이다. D latch는 clock이 들어와 있을 때 input d값에대한 output q를 출력한다. |

1. D Flip-Flop

|  |
| --- |
| C:\Users\Administrator\Desktop\123.jpg |
| 위 그림은 D flip-flop 의 symbol이다. D flip-flop는 clock이 rising edge일때 input d값에 대한 output q를 출력한다. |

1. enabled D flip-flop

|  |
| --- |
|  |
| 위 그림은 enabled D flip-flop 의 symbol이다. enabled D flip-flop은 D flip-flop 앞에 2-to-1 mx를 추가하여 en에 따라서 input값의 사용여부를 결정한다. |

1. Resettable D Flip-Flop

|  |
| --- |
|  |
| 위 그림은 D flip-flop에서 reset기능이 추가된 Resettable D flip-flop 이다. Active low일 때 동작하도록 설계하였다. Reset이 1로 올라가야지만 입력 값에 대한 결과값을 출력한다. |

1. Synchronous Set/Resettable D Flip-Flop

|  |
| --- |
|  |
| 위 그림은 D flip-flop에서 reset, set기능이 추가된 Synchronous Set/Resettable D flip-flop 이다. Set과 reset은 active low에 동작하도록 설계하였고, reset신호가 set신호보다 우선순위를 가지도록 설계하였다. |

1. 32-bits Register

|  |
| --- |
|  |
| 위 그림은 32bit Register의 symbol 이다. Input과 output 모두 32bit 이며, D Flip Flop 32개로 구성되어있다. Clk은 모두 동일한 신호로 들어간다. 즉, N개의 Flip Flop 이 instance되어있는 circuit이 N-bit Register이다. |

1. Nand Gate 4개를 이용하여 D latch를 구현하는 방법

|  |
| --- |
|  |
|  |
| 위의 첫번째 그림은 실습 시간에 구현한 방법의 D-latch 의 schematic이고, 두번째 그림은 NAND Gate 를 4개만 사용하여 구현한 D-latch 의 schematic 이다. CLK이 0이면 D의 값의 상관없이 Q는 이전Q값, Q’는 이전 Q` 가 나온다. CLK이 1로 뛸 때 D의 값이 0 이면, Q는 0, Q’는 1이 나오고, CLK 이 1일 때 D가 1이면 Q는 1 Q’는 0 이 나온다. |

1. enabled D flip-flop을 다른 방법으로 구현하는 방법

|  |  |
| --- | --- |
|  |  |
| 위의 왼쪽 그림은 실습시간 구현한 enabled D-flip-flop의 schematic이고, 위의 오른쪽 그림은 다른 방법으로 구현한 enabled D-flip-flop이다. 실습 시간에 사용한 방법처럼 2-to-1 mux를 사용하지 않았고, clk 자체에 enable 신호를 AND Gate 로 추가하여 설계하였다. Clk이 0이거나 enable 신호가 0이라면 clk자체가 아예 들어가지 않고, clk이 1일때에 enabled가 1로 들어오면 d의 값이 q의 값으로 가게 되고, clk이 1일 때 enable 이 0이라면 clk자체가 들어가지 않으므로, q값은 나오지 않는다. 따라서 enable이 1일 경우에 clk에 값에 따라서 flip-flop이 동작하도록 설계하였다. | |

1. **Traffic Light Controller with/without Left Turn Signals**
2. Moore FSM

|  |
| --- |
| <Moore FSM> |
|  |
| 위 그림은 출력이 현재 상태에 의해서만 결정이 되는 Moore FSM이다.  다른 입력에 대해서 영향을 따로 받지 않는다. |

1. Traffic Light Controller

신호등을 제어하는 logic을 구현하는 Traffic Light Controller 설계하기 위해서는 몇가지 규칙이 있다.

첫째, 신호등 LA는 Academic.ave(가로 길)의 차량 통행을 제어하는 신호등 이다.

둘째, 신호등 LB는 Bravado.blvd(세로 길)의 차량 통행을 제어하는 신호등 이다.

셋째, Traffic Light Controller 는 신호등이 시간에 따라서 자동으로 불의 색깔의 바뀌는 신호등이 아니라 자동차가 차로에 있을 때 신호등은 Green 으로 유지가 되고 자동차가 차로에 없어지면 신호등이 Green 에서 Yellow 을 거쳐 Red 로 바뀌는 신호등이다.

만약 LA가 Green이나 Yellow 면 LB는 Red이고 LA가 Red면 LB는 Green이나 Yellow이다.

넷째, 차량이 차도에 있는지 없는지를 확인하기 위하여 Academic.ave에는 TA를

Bravado.blvd에는 TB를 설치한다.

1. Traffic Light Controller with Left Turn Signals

앞서 구현한 Traffic Light Controller 에서 좌회전 기능을 추가한 Traffic Light Controller with Left Turn Signals 이다. Traffic Light Controller with Left Turn Signals 를 설계하기 위해서는 몇가지 규칙이 있다.

첫째, 신호등 LA는 Academic ave(가로 길)의 차량 통행을 제어하는 신호등 이다.

둘째, 신호등 LB는 Bravado blvd(세로 길)의 차량 통행을 제어하는 신호등 이다.

셋째, 신호등은 거리에 차가 있을 때 신호등은 Green 또는 Left 가 되며, 없을 때는 Red 가 된다.

넷째, 차량이 차도에 있는지 없는지를 확인하기 위하여 Academic ave(가로길)에는 TA, TAL,를 Bravado blvd(세로길)에는 TB, TBL를 설치한다. TA, TB는 직진 차량에 대한 감지 센서고, TAL, TBL은 좌회전 차량에 대한 감지센서이다.

다섯째, 신호등은 차도에 차가 없을 때, Green 에서 Yellow을 거쳐 Left로 변한다.

신호등은 차도에 차가 없을 때, Left에서 Yellow을 거쳐 Red으로 변한다.

신호등은 좌회전하는 차가 없어도 Green에서 Red로 변하는게 아닌 Left 로 우선적으로 바뀌어야 한다.

만약 신호등 LA가 Green, Yellow 또는 Left일 경우에 LB는 Red 이어야 하고, 반대의 경우에도 적용 되어야 한다.

1. 설계 세부사항
2. **Latch & flip-flop design with/without reset/set**
3. SR Latch

<SR- Latch의 truth table>

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | Output | |
| S | R | Q | Q’ |
| 0 | 0 | 이전Q | 이전Q` |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

|  |
| --- |
|  |
| <SR Latch의 Schematic> |

1. D Latch

<D-latch의 truth table>

|  |  |
| --- | --- |
| Input | Output |
| CLK | Q |
| 0 | 이전 Q |
| 1 | D |

|  |
| --- |
|  |
| <D Latch의 Schematic> |

1. D Flip-Flop

<D-Flip-Flop의 truth table>

|  |  |
| --- | --- |
| Input | Output |
| CLK | Q |
| Rising edge | D |
| Other case | 이전 Q |

|  |
| --- |
|  |
| <D Flip-Flop의 Schematic> |

1. Enabled D Flip-Flop

<Enabled D Flip-Flop의 truth table>

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | | Output |
| EN | CLK | D | Q |
| 0 | X | X | 이전Q |
| 1 | Rising edge | D | D |

|  |
| --- |
|  |
| <Enabled D Flip-Flop의 Schematic> |

1. Resettable D Flip-Flop

<Resettable D Flip-Flop의 truth table>

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | | Output |
| R | D | CLK | Q |
| 0 | X | X | 0 |
| 1 | 0 | Rising edge | 0 |
| 1 | 1 | Rising edge | 1 |
| 1 | X | Falling edge  Or 0 or 1 | 이전 Q |

|  |
| --- |
|  |
| <Resettable D Flip-Flop의 Schematic> |

1. Synchronous Set/Resettable D Flop-Flop

< Synchronous Set/Resettable D Flip-Flop의 truth table>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | | | | Output |
| R | S | CLK | D | Q |
| 0 | X | X | X | 0 |
| 1 | 0 | X | X | 1 |
| 1 | 1 | Rising edge | D | D |

|  |
| --- |
|  |
| <Synchronous Set/Resettable D Flip-Flop의 Schematic> |

1. Register

|  |
| --- |
|  |
| <32bits Register의 Schematic> |

32-bits Register는 위의 그림처럼 8bits register 4개를 instance하여 설계하였다.

1. Async/Sync Set/Resettable D Flip-Flop

Async/SyncSet/Resettable D Flip-Flop 을 설계하는데 있어서 앞선 structural implementation 방법이 아닌, behavioral implementation 방법을 통하여 두 FlipFlop 의 차이점을 확인하였다. sync에서는 clk이 rising edge 일 때에만, reset과 set값이 나오지만, async에서는 clk이 rising edge일 때 뿐만 아니라, set이나, reset이 falling edge일때도 reset과 set값이 된다.

1. **Traffic Light Controller with/without Left Turn Signals**
2. Traffic Light Controller

<Define states>

|  |  |
| --- | --- |
| S0 | 00 |
| S1 | 01 |
| S2 | 10 |
| S3 | 11 |
| G(Green) | 00 |
| Y(Yellow) | 01 |
| R(Red) | 10 |

|  |  |  |
| --- | --- | --- |
|  | LA | LB |
| S0 | G | R |
| S1 | Y | R |
| S2 | R | G |
| S3 | R | Y |

|  |
| --- |
| <state diagram> |
|  |
| 위 그림은 Traffic Light Controller의 state diagram이다. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Current | | Input | | Next | |
| Q1 | Q0 | TA | TB | D1 | D0 |
| S0 | 0 | 0 | 1 | X | 0 | 0 |
| 0 | 0 | 0 | X | 0 | 1 |
| S1 | 0 | 1 | X | X | 1 | 0 |
| S2 | 1 | 0 | X | 1 | 1 | 0 |
| 1 | 0 | X | 0 | 1 | 1 |
| S3 | 1 | 1 | X | X | 0 | 0 |

위의 테이블들을 이용하여, ns logic과 o logic을 정리하면 아래와 같은 식을 얻을 수 있다.

<ns logic>

D1 = Q1^Q0

D0 = Q1’Q0’TA’ + Q1Q0’TB’

<o logic>

LA1=Q1

LA0=Q1’Q0

LB1=Q1’

LB2=Q1Q0

1. Traffic Light Controller with Left Turn Signals

<Define states>

|  |  |
| --- | --- |
| S0 | 000 |
| S1 | 001 |
| S2 | 010 |
| S3 | 011 |
| S4 | 100 |
| S5 | 101 |
| S6 | 110 |
| S7 | 111 |
| G(Green) | 00 |
| Y(Yellow) | 01 |
| R(Red) | 11 |
| L(Left) | 10 |

|  |
| --- |
| <state diagram> |
|  |
| 위 그림은 Traffic Light Controller with Left Turn Signals 의 state diagram이다. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Current | | | Input | | | | Next | | |
| Q2 | Q1 | Q0 | TA | TB | TAL | TBL | D2 | D1 | D0 |
| S0 | 0 | 0 | 0 | 1 | X | X | X | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | X | X | X | 0 | 0 | 1 |
| S1 | 0 | 0 | 1 | X | X | X | X | 0 | 1 | 0 |
| S2 | 0 | 1 | 0 | X | X | 1 | X | 0 | 1 | 0 |
| 0 | 1 | 0 | X | X | 0 | X | 0 | 1 | 1 |
| S3 | 0 | 1 | 1 | X | X | X | X | 1 | 0 | 0 |
| S4 | 1 | 0 | 0 | X | 1 | X | X | 1 | 0 | 0 |
| 1 | 0 | 0 | X | 0 | X | X | 1 | 0 | 1 |
| S5 | 1 | 0 | 1 | X | X | X | X | 1 | 1 | 0 |
| S6 | 1 | 1 | 0 | X | X | X | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | X | X | X | 0 | 1 | 1 | 1 |
| S7 | 1 | 1 | 1 | X | X | X | X | 0 | 0 | 0 |

위의 테이블들을 이용하여, ns logic과 o logic을 정리하면 아래와 같은 식을 얻을 수 있다.

<ns logic>

D2 = Q2’Q1Q0 + Q2Q1’ + Q2Q1Q0‘

D1 = Q2’Q1’Q0 + Q1Q0’ + Q2Q1’Q1

D0 = Q2’Q1’Q0’TA’ + Q2’Q1Q0’TAL’ + Q2Q1’Q0’TB’ + Q2Q1Q0’TBL’

<o logic>

LA1 = Q1Q0’ + Q2

LA0 = Q0 + Q2

LB1 = Q2’ + Q1Q0’

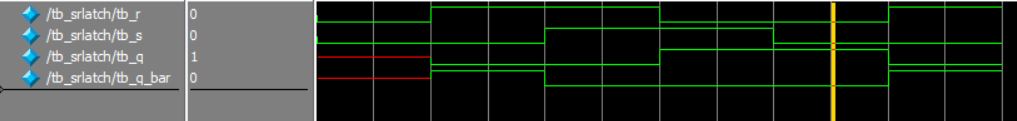
LB0 = Q2’ + Q0

#. ns logic 과 o logic 을 behavior design 과 structural design 으로 구현했을 때의 크기 비교

|  |  |
| --- | --- |
| C:\Users\장한별 (Stanley)\Desktop\2.be.flow.JPG | C:\Users\장한별 (Stanley)\Desktop\2.st.flow.JPG |

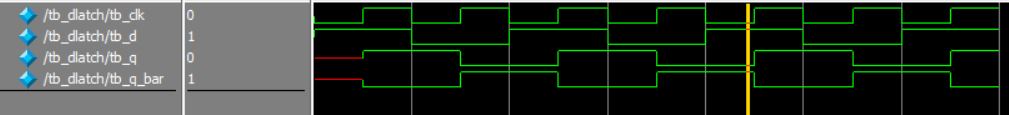
위 두 그림들은 Traffic Light Controller with Left Signals 의 behavior design 의 FlowSummary와 structural design 의 FlowSummary 이다. 보는 바와 같이 Total logic elements 는 9로, Total pins 는 10 으로 같다는 것을 확인할 수 있다. 따라서 크기는 두가지 방법을 구현해도 같다.

1. 설계 검증 및 실험 결과
2. **Latch & flip-flop design with/without reset/set**
   1. 시뮬레이션 결과



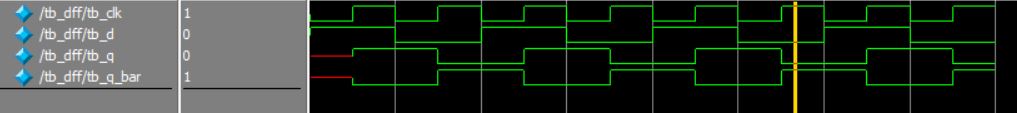
**<SR-Latch 의 WaveForm>**

위 그림은 SR-Latch 의 WaveForm 이다. 처음 R 과 S가 0일 때는 이전 값이 없으므로 Unknown 으로 나오고 있다. 그 이후, 진리표대로 출력이 잘 되었음을 확인 할 수 있다.

****

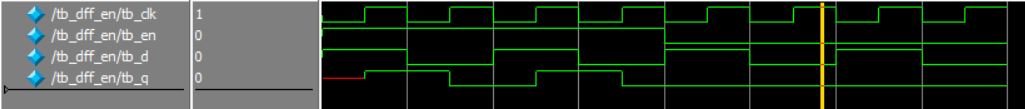
**<D-latch 의 WaveForm>**

위 그림은 D-Latch 의 WaveForm 이다. Clk이 1일 때, d의 값이 0 이면 q는 0, d의 값이 1이면 q=1로 출력되는 것을 확인할 수 있다. Clk이 0일 때, q는 이전 q를 유지하는 것을 확인할 수 있다.

****

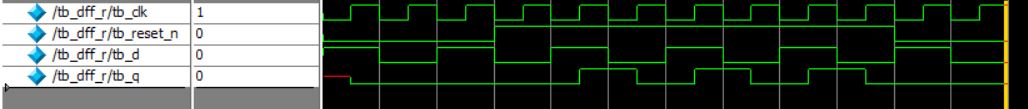
**<D-Flip Flop 의 WaveForm>**

위 그림은 D FlipFlop의 WaveForm 이다. Clk이 Rising edge 일 때, d의값이 0이면 q는 0, d의 값이 1이면 q=1 으로 출력되는 것을 확인할 수 있다. 이외의 경우에는 q는 이전 값을 유지한다.

****

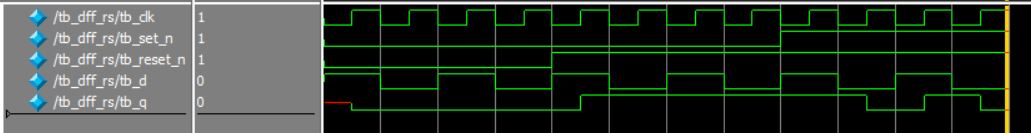
**<enable D-Flip Flop 의 WaveForm>**

위 그림은 Enable D FlipFlop 의 WaveForm 이다. en 값이 1일 때, 정상적으로 작동을 하다가 en값이 0일 때, 이전 q값을 계속 유지하는 것을 확인할 수 있다.

****

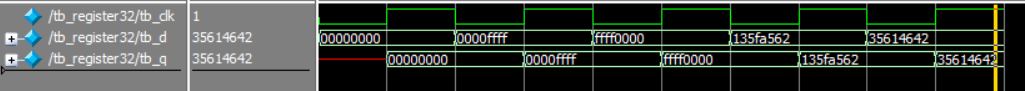
**<Resettable D-Flip Flop 의 WaveForm>**

위 그림은 Resettable D FlipFlop의 WaveForm 이다. 처음 reset 값이 0일 때, 작동을 하지 않다가 reset을 1로 올려주면, clock이 Rising edge 일때 d의 값이 1이면 q의 값이 1이되고, d의 값이 0이면 q의 값이 0이 되는 것을 확인할 수 있다.

****

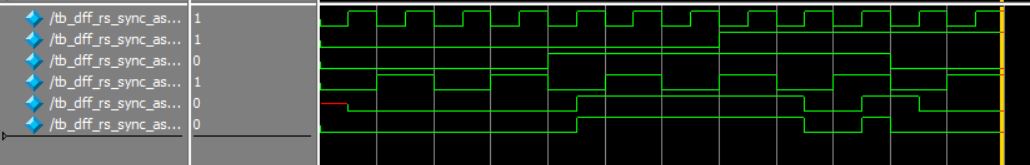
**<Synchronous Set/Resettable D-Flip Flop 의 WaveForm >**

위 그림은 Synchronous Set/Resettable D FlipFlop 의 WaveForm 이다. Reset 이 set보다 우선 순위에 있고, reset 이 0이면, q는 계속 0을 유지하다가 reset을 1로 올려주면, q는 1을 계속 유지한다. 이때, set을 1로 올려주면 clk이 rising edge일 때, q가 1로 올라가고, 다른 경우에는 q가 0으로 떨어지는 것을 확인할 수 있다.

****

**<32-bits Register 의 WaveForm>**

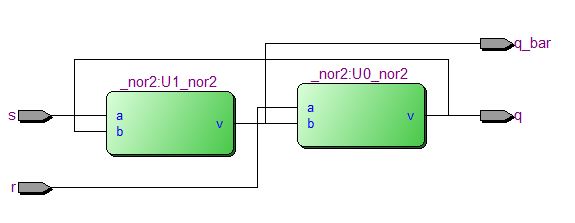
위 그림은 32-bits Register 의 WaveForm 이다. Clk이 Rising edge 일 때, input d 에 따라서 output q를 d값으로 출력됨을 확인할 수 있다.



**<Async/Sync Set/Resettable D-Flip Flop 의 WaveForm >**

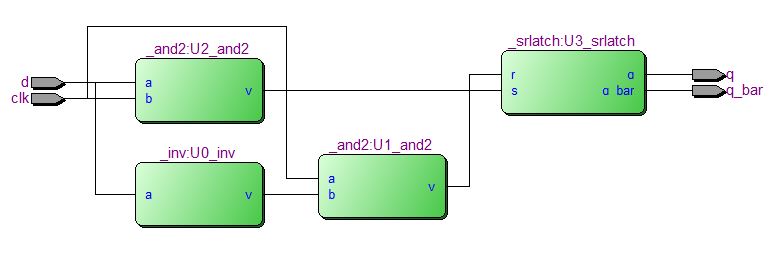
위 그림은 Async/Sync Set/Resettable D FlipFlop의 WaveForm 이다. reset 신호가 1에서 0으로 떨어질 때, async는 바로 0으로 set 이 되고, Sync는 clock이 rising edge 일때, 0으로 set 이 됨을 확인할 수 있다.

* 1. 합성(synthesis) 결과



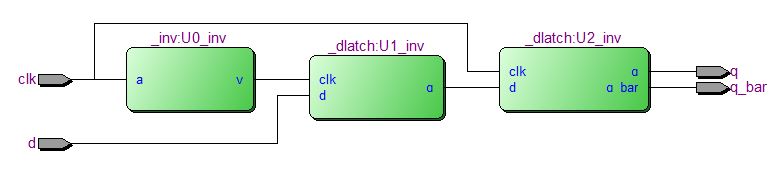
**<SR-Latch 의 RTL Map Viewer>**

위 그림은 SR-Latch 의 RTL Map Viewer 이다. 2input NOR Gate 2개를 instance 하여 설계되었음을 확인할 수 있다.

****

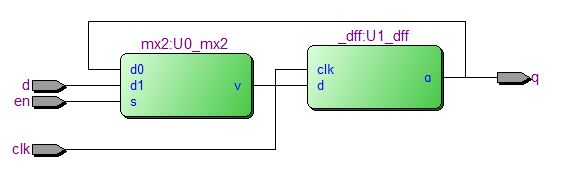
**<D-Latch 의 RTL Map Viewer>**

위 그림은 D-Latch 의 RTL Map Viewer 이다. 2input AND Gate 2개와 inverter, SR Latch 를 instance 하여 설계되었음을 확인할 수 있다.



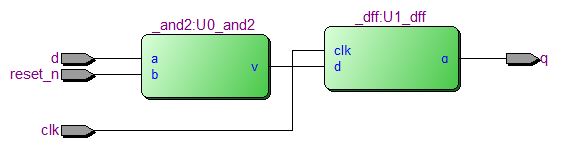
**<D-Flip Flop 의 RTL Map Viewer>**

위 그림은 D FlipFlop의 RTL Map Viewer이다. D Latch 2개와 inverter 를 instance 하여 설계되었음을 확인할 수 있다.

****

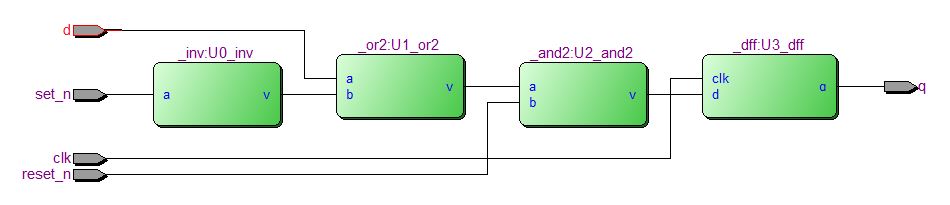
**<enable D-Flip Flop 의 RTL Map Viewer>**

위 그림은 Enable D FlipFlop 의 RTL MapViewer 이다. 2to1 mx 와 D FlipFlop을 instance 하여 설계되었음을 확인할 수 있다.

****

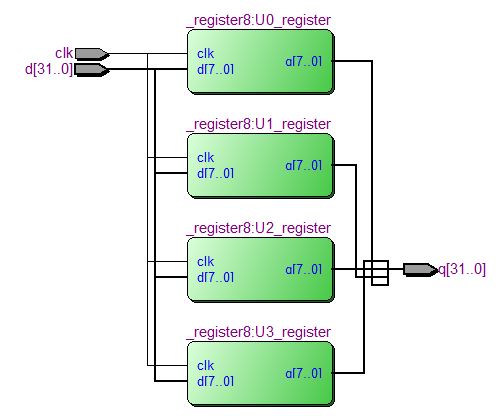
**<Resettable D-Flip Flop 의 RTL Map Viewer>**

위 그림은 Resettable D FlipFlop의 RTL Map Viewer 이다. 2input AND Gate 와 D FlipFlop을 instance하여 설계되었음을 확인할 수 있다.



**< Set/Resettable D-Flip Flop 의 RTL Map Viewer >**

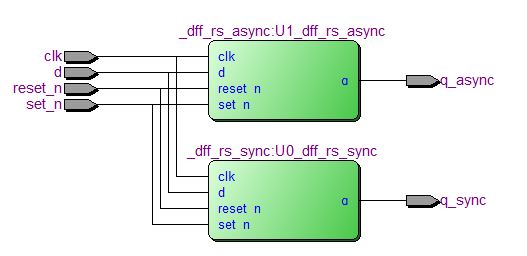
위 그림은 Synchronous Set/Resettable D FlipFlop의 RTL Map Viewer 이다. 2input AND Gate 와 2input OR Gate, D Flipflop 을 instance 하여 설계되었음 확인할 수 있다.

****

**<32-bitsRegister 의 RTL Map Viewer>**

.

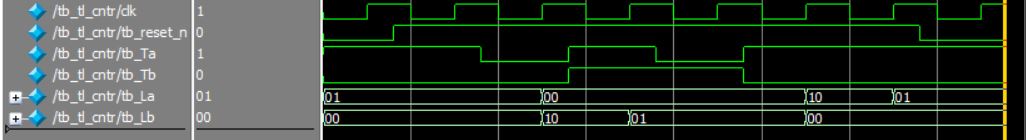
위 그림은 32-bits Register 의 RTL Map Viewer이다. 8-bits Register 4개를 instance 하여 설계되었음을 확인할 수 있다.



**<Async/Sync Set/Resettable D-Flip Flop 의 RTL Map Viewer >**

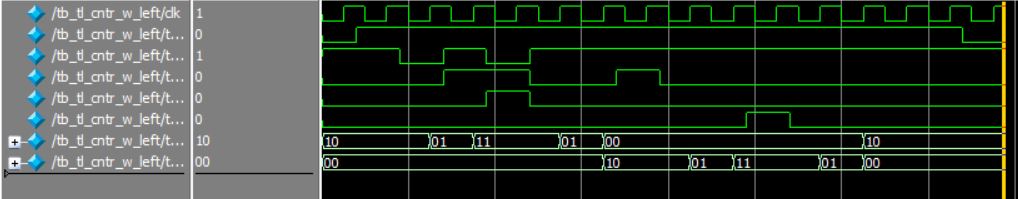
위 그림은 Async/Sync Set/Resettable D-Flip Flop 의 RTL Map Viewer 이다. Async 와 Sync 를 instance 하여 설계되었음을 확인할 수 있다.

1. **Traffic Light Controller with/without Left Turn Signals**
   1. 시뮬레이션 결과

.

**<Traffic Light Controller 의 WaveForm>**

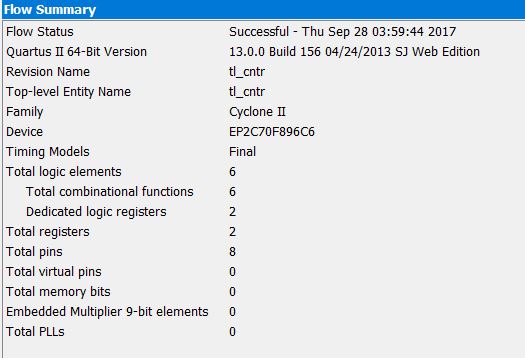
위 그림은 Traffic Light Controller 의 WaveForm 이다. 첨부한 State diagram 그림 순서대로 input Ta, Tb 가 차례대로 들어가고 이에 따른 output La, Lb 의 값이 정확히 나온 것을 확인할 수 있다.



**<Traffic Light Controller with Left Turn Signals 의 WaveForm>**

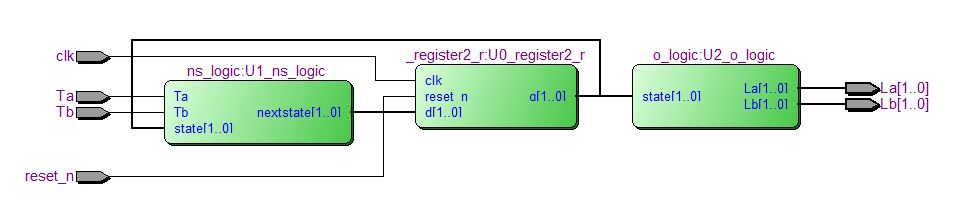
위 그림은 Traffic Light Controller with Left Turn Signals 이다. 첨부한 state diagram 그림 순서대로 input값 Ta, Tb, Tal, Tbl 가 차례대로 들어가고 이에 따른 output La, Lb의 값이 정확하게 나온 것을 확인할 수 있다.

* 1. 합성(synthesis) 결과



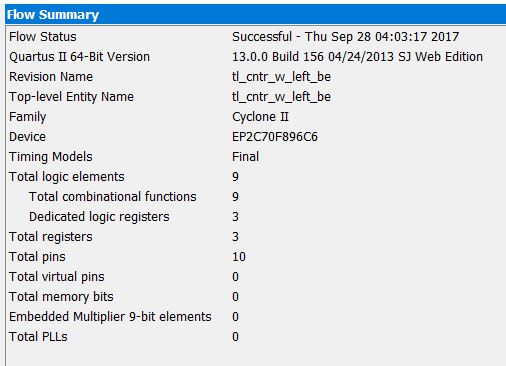
**<Traffic Light Controller 의 FlowSummary>**

위 그림은 Traffic Light Controller 의 FlowSummary 이다. Total logic elements 는 6, total pins 은 8 임을 확인할 수 있다.



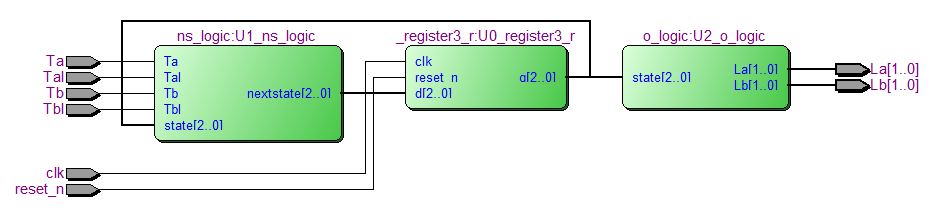
**<Traffic Light Controller 의 RTL Map Viewer>**

위그림은 Traffic Light Controller 의 RTL Map Viewer 이다. ns\_logic과 o\_logic 그리고 register2 를 instance 하여 설계되었음을 확인할 수 있다.



**<Traffic Light Controller with Left Turn Signals 의 FlowSummary>**

위 그림은 Traffic Light Controller with Left Turn Signals 의 FlowSummary 이다. Total logic elements 는 9, Total pins 는 10 임을 확인할 수 있다.



**<Traffic Light Controller with Left Turn Signals 의 RTL Map Viewer>**

위 그림은 Traffic Light Controller with Left Turn Signals 의 RTL Map Viewer 이다. ns\_logic 과 o\_logic 그리고 register3\_r 을 instance 하여 설계되었음을 확인할 수 있다.

1. 고찰 및 결론
   1. 고찰
      1. **Latch & flip-flop design with/without reset/set**

이번 실습 시간에는 각종 Latch 와 Flip Flop을 구현하고, 설계했다. Latch와 FlipFlop 의 공통점은 둘 다 기본적인 기억 소자 장치이다. 신호를 받다가 중간에 끊기게 되더라 도 그 전의 값들을 저장하고 있다. 차이점은 CLK이 Rising edge 일때 FlipFlop은 input 값에 따라 output값을 변화시키는 특징이 있다.

여러가지 실습 중 D FlipFlop을 synchronous set/resettable과Asynchronous set/resettable 을 비교해서 진행했다. 먼저 Synchronous는 Clock에 따라서 set과 reset 을 설정하고, Asynchronous는 clock 과 관계없이 set과 reset을 설정한다. 즉, Synchronous set/resettable D Flipflop은 clock 이 rising edge를 하지 않으면 set과 reset을 설정해도 결과값으로 영향을 미치지 않고 clk이 rising edge가 될 때 reset과 set 값의 영향을 받는다. 반면에 Asynchronous set/resettable D flipflop 에서는 clock 이 설정되기 전 set이나 reset이 falling edge 일 때, set 또는 reset이 결과값이 영향을 준다는 것을 알았다.

* + 1. **Traffic Light Controller with/without Left Turn Signals**

이번 실습 시간을 통해 Traffic Light Controller를 직접 설계하고 구현하였다. FSM 을 이용하고 설계하고 diagram을 그리고 state transition table, output table 을 작성하고 이에 따른 Boolean equation을 작성하였다. 이를 통해 FSM 을 설계하는데 더욱 발전한 느낌을 받았다. 또한 ns logic 과 o logic을 structural design 방식과 behavior design 방식 2가지 방식으로 구현 가능하다는 점을 배웠다. 그리고 실제로 구현을 해보니 사이즈 자체는 같은 것으로 확인이 되었다. 기존에 짜던 방식과 비슷한 structural design 방식이 더 간결하고 쉽게 와닿았다. 앞으로 어떤 방식이 더욱 유용하게 쓰일지 확인해 봐야할 것 같다.

* 1. 결론
     + 1. **Latch & flip-flop design with/without reset/set**

Latch와 FlipFlop은 기억 소자로, Latch 는 Clock이 0일 때, FlipFlop은 clock이 Rising edge가 아닌 경우에 이전 Q값을 기억한다. FlipFlop은 Latch를 이용해서 구현할 수 있다. Latch는 clock이 1이면 D값을 Q값으로 내보내기 때문에, 값을 오래 저장하지 못하고 자주 바뀔 수 있다. 하지만 FlipFlop은 clock이 rising edge 일때만 값이 바뀌기 때문에 이전 값을 한 cycle 동안 저장할 수 있다. enabled D FlipFlop은 Clock 이외에 en signal이 flipflop의 동작을 제어할 수 있고, set/resettable D flipflop은 flipflop이 처음 z값을 가질 때 값을 초기화하여 제대로 값이 전달될 수 있도록 한다. Sync/Async set/resettable D FlipFlop은 set과 reset동작을 sync로 하는지 async로 하는지에 따라 다른데 sync set/resettable D Flipflop은 reset과 set 동 작이 clock의 rising edge에서 동작되고 Async set/resettable D Flipflop은 clock과 관계없이 reset/set signal 이 0이 되는 순간 그 기능을 한다.

* + - 1. **Traffic Light Controller with/without Left Turn Signals**

Next state logic(ns logic) 과 output state logic(o logic)을 gate level로 structural design 과 case 문을 이용한 behavior design 으로 구현하였다. 서로 다른 방식이지만 같은 결과를 가졌다. 그렇지만 회로가 더 커지고 복잡하다면 behavior design 의 회로가 더 많은 논리 요소가 필요하게 된다. gate level 의 structural design 의 경우 출력 값을 일정한 경우와 공식으로 산출하는 반면, behavior design 의 경우 흐름에 따른 경우의 수로 출력 값을 산출하기 때문에 더 많은 회로가 필요하게 된다.

1. 참고문헌

공진흥 / 컴퓨터공학기초실험2 / 새빛관303호(광운대학교) / 2017년.

이준환 / 디지털논리회로2 / 참빛관B101호(광운대학교) / 2017년.