11주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231523 이름: 김민정

**1. RS Flip-Flop (with Simulation)**

RS FF(이제부터 Flip Flop을 FF로 서술하겠다.) R이 1이면 Q가 0으로 set이 되고 S가 1이면 Q가 1로 세팅 된다. 만약 R, S가 모두 0이라면 이전의 값이 유지되는 특성을 가진다. 아래는 RS FF를 NOR gate와 NAND gate로 나타내는 과정을 서술한 것이다.

1) truth table

아래 표는 clock bit가 모두 1이라는 것을 가정하여 작성한 표이다. 만약 clock bit가 0이면 Q는 이전의 값을 계속 유지한다. 또한 R, S가 모두 0이어도 Q는 이전 값을 유지한다. R, S가 모두 1인 상황은 논리적 모순이 발생하므로 이는 설정하지 않는다. (이론에서는 edge에서 이러한 논리과정이 일어나야 하지만, 아직 이러한 코드 작성법을 배우지 않았기에 static state인 low나 high 과정에 적용하겠다. 이후 설명할 D FF도 동일하게 적용된다.)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 입력 순서 | R | S | Q | ~Q |
| 1 | 0 | 1 | 1 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 1 |
| 5 | 1 | 0 | 0 | 1 |
| 6 | 1 | 1 |  | |

2) code

가) NAND RS FF

|  |  |
| --- | --- |
| Design code | Tesbench code |
| `timescale 1ns / 1ps  module RSff\_NAND(  input s, r, clk,  output q, qbar  );    assign q = ~(~(s&clk) & qbar);  assign qbar = ~(~(r & clk) & q);  endmodule | `timescale 1ns / 1ps  module RSff\_NAND\_tb;  reg clk, s, r;  wire q, qbar;  RSff\_NAND u\_test(  .clk (clk ),  .s (s ),  .r (r ),    .q (q ),  .qbar (qbar )  );  initial begin  clk = 1'b0;  r = 1'b0;  s = 1'b1;  end    always clk = #25 ~clk;  always r = #20 ~r;  always s = #30 ~s;    initial begin  #1000  $finish;  end  endmodule |

나) NOR RS FF

|  |  |
| --- | --- |
| Design code | Tesbench code |
| `timescale 1ns / 1ps  module RSff\_NOR(  input s, r, clk,  output q, qbar  );    assign q = ~((r&clk) | qbar);  assign qbar = ~((s & clk) | q);  endmodule | `timescale 1ns / 1ps  module RSff\_NOR\_tb;  reg clk, s, r;  wire q, qbar;  RSff\_NOR u\_test(  .clk (clk ),  .s (s ),  .r (r ),    .q (q ),  .qbar (qbar )  );  initial clk = 1'b0;  initial r=1'b0;  initial s=1'b1;    always clk = #20 ~clk;  always r = #50 ~r;  always s = #100 ~s;    initial begin  #1000  $finish;  end  endmodule |

3) simulation

|  |
| --- |
| NAND RS FF simulation |
| 스크린샷, 입력 장치, 컴퓨터, 전자 기기이(가) 표시된 사진  자동 생성된 설명 |

|  |
| --- |
| NOR RS FF simulation |
| 스크린샷, 사각형이(가) 표시된 사진  자동 생성된 설명 |

**2. D Flip-Flop (with Simulation)**

D FF(이제부터 Flip Flop을 FF로 서술하겠다.)는 D input에 따라 output이 정해지는 FF이다. 즉 D가 1이면 output인 Q는 1이 되고, D가 0이면 Q는 0이 된다. 이도 clk이 1일때에만 작동된다.

1) truth table

|  |  |  |  |
| --- | --- | --- | --- |
| 입력순서 | D | Q | ~Q |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 1 | 1 | 0 |
| 6 | 1 | 1 | 0 |

2) code

|  |  |
| --- | --- |
| Design code | Tesbench code |
| `timescale 1ns / 1ps  module Dff(  input clk, d,  output q, qbar  );  assign q = ~((~d &clk)|qbar);  assign qbar = ~((d & clk) |q);  endmodule | `timescale 1ns / 1ps  module Dff\_tb;  reg clk, d;  wire q, qbar;  Dff u\_test(  .clk (clk ),  .d (d ),    .q (q ),  .qbar (qbar )  );  initial clk = 1'b0;  initial d=1'b0;    always clk = #20 ~clk;  always d = #50 ~d;    initial begin  #500  $finish;  end  endmodule |

3) simulation

|  |
| --- |
| simulation |
| 스크린샷이(가) 표시된 사진  자동 생성된 설명 |

**3. 결과 검토**

RS FF과 D FF을 제작하면서 clock의 유무의 따른 장치 작동 코드를 작성해볼 수 있었다. 하지만 이러한 코드는 edge를 고려하지 않기 때문에 완전한 FF 구현이라고 하기엔 어렵다. 이를 반영한 RS FF을 4번 문항에서 제작해보면서 edge를 고려한 코드 작성을 실습해보도록 하자.

**4. edge를 고려한 RS Flip Flop(추가 이론 조사 및 작성)**

Edge를 고려한 RS FF는 always @(posedge clock) begin를 통해 구현할 수 있다. Posedge는 rising edge일 때 아래의 코드를 실행시킨다는 뜻이다. Clock이 작동할 때의 R, S의 작동은 위의 NAND와 NOR으로 구현한 것과 달리 조건문을 통해 구현했다. 아래는 edge를 고려한 RS FF의 code와 simulation 결과이다.

1) code

|  |  |
| --- | --- |
| Design code | Tesbench code |
| `timescale 1ns / 1ps  module RSff(  input clock, s, r,  output q, qbar  );  reg q;  assign qbar = ~q;  always @(posedge clock) begin  if(s&&r)  q <= 1'bx;  else if(s&&(~r))  q <= 1'b1;  else if((~s)&&r)  q <= 1'b0;  end  endmodule | `timescale 1ns / 1ps  module RSff\_tb;  reg clock, s, r;  wire q, qbar;  RSff u\_test(  .clock (clock ),  .s (s ),  .r (r ),    .q (q ),  .qbar (qbar )  );  initial clock = 1'b0;  initial r=1'b0;  initial s=1'b1;    always clock = #20 ~clock;  always r = #50 ~r;  always s = #100 ~s;    initial begin  #1000  $finish;  end  endmodule |

2) simulation

|  |
| --- |
| simulation |
|  |

Simulation 결과를 통해 rising edge가 있을 때만 작동함을 확인할 수 있다. R, S input이 모두 1일 때는 output을 0으로 설정했다.