14주차 결과보고서

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

**1. 1101 overlapping Sequence Detector ( in Moore machine)**

Overlapping 방식의 1101 sequence detector을 제작해보고자 한다. 여기서 Overlapping이란 sequence detector에서 특정 조건을 만족하는 sequence가 계속 나올 수 있음을 고려하여 제작하는 방식이다. Overlapping 장치는 특정 조건, 즉 패턴이 겹쳐서 나타날 때마다 이를 계속 인식하고 검출할 수 있어야 한다.

이제 1101 overlapping sequence을 Moore machine으로 제작해보자. 먼저 state diagram을 그려 기계장치 작동장치 흐름을 design 해보자. 아래는 state diagram & table과 이를 통해 작성한 Verilog code이다. 마지막 사진은 해당 코드의 behavioral simulation이다

1) state diagram and state table

그림, 스케치, 라인 아트, 클립아트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
| Q(present) | Q(next) | | output |
| X=0 | X=1 |
| A | A | B | 0 |
| B | A | C | 0 |
| C | D | C | 0 |
| D | A | E | 0 |
| E | A | C | 1 |

표에서 Q(present)는 현재 state, Q(next)는 다음 state를 의미한다. 마지막으로 x는 input을 의미한다. 이후 서술하는 state table도 위와 같은 setting을 가진다.(이후부터는 이에 대한 서술을 하지 않고 위의 설정을 고정하여 진행할 것이다.)

2) Verilog source

|  |  |
| --- | --- |
| Design code | Testbench code |
| `timescale 1ns / 1ps  module moore(  input clk, reset,  input in,  output out,  output [3:0] Q  );    reg out;  reg A,B,C,D,E;  reg [3:0] Q;  initial begin  A = 1'b1;  B = 1'b0;  C = 1'b0;  D = 1'b0;  E = 1'b0;  out = 1'b0;  Q=1'b0000;  end    always @(posedge clk or posedge reset)  begin  if(reset)  begin  A <= 1'b1;  B <= 1'b0;  C <= 1'b0;  D <= 1'b0;  E <= 1'b0;  out <= 1'b0;  end  else if(A==1'b1)  begin  if(in)  begin  B <= 1'b1;  A <= 1'b0;  end  else  begin  A <= 1'b1;  end  out <= 1'b0;  end  else if(B==1'b1)  begin  if(in)  begin  B<=1'b0;  C<=1'b1;  end  else  begin  B<=1'b0;  A<=1'b1;  end  out<=1'b0;  end  else if(C==1'b1)  begin  if(in)  begin  C<=1'b1;  end  else  begin  C<=1'b0;  D<=1'b1;  end  out<=1'b0;  end  else if(D==1'b1)  begin  if(in)  begin  D<=1'b0;  E<=1'b1;  end  else  begin  D<=1'b0;  A<=1'b1;  end  out<=1'b0;  end  else if(E==1'b1)  begin  if(in)  begin  E<=1'b0;  C<=1'b1;  end  else  begin  A<=1'b1;  E<=1'b0;  end  out<=1'b1;  end  end  always @(posedge clk) begin  if(reset) begin  Q = 4'b0000;  end  else begin  Q[0] <= Q[1];  Q[1] <= Q[2];  Q[2] <= Q[3];  Q[3] <= in;  end  end    endmodule | `timescale 1ns / 1ps  module moore\_tb;  reg clk,reset,in;  wire out,A,B,C,D,E;  moore u\_test(  .clk ( clk),  .reset ( reset),  .in ( in),    .out (out),  .A (A),  .B (B),  .C (C),  .D (D ),  .E (E )    );  initial begin  reset = 1'b1;  in=1'b1;  clk = 1'b1;  end  always reset = #200 ~reset;  always in= #20 $urandom%2;  always clk = #10 ~clk;  initial begin  #1000  $finish;  end  endmodule |

3) Simulation

스크린샷이(가) 표시된 사진

자동 생성된 설명

위 시뮬레이션 사진에서 1101 입력 후 다음 턴 output에서 1이 나옴을 알 수 있다.

**2. 10101 overlapping Sequence Detector ( in Mealy machine)**

이제 mealy machine을 통해 10101 sequence detector을 제작해보겠다. Mealy machine은 Moore machine가 다르게 input이 들어가면 해당 turn에 바로 output이 1이 검출되어야 한다.

1) state diagram and state table

그림, 스케치, 라인 아트, 클립아트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q(present) | Q(next) | | output | |
| X=0 | X=1 | X=0 | X=1 |
| A | A | B | 0 | 0 |
| B | C | B | 0 | 0 |
| C | A | D | 0 | 0 |
| D | E | B | 0 | 0 |
| E | A | D | 0 | 1 |

2) Verilog source

|  |  |
| --- | --- |
| Design code | Testbench code |
| `timescale 1ns / 1ps  module mealy(  input clk, reset,  input in,  output out,  output [4:0] Q  );    reg out;  reg A,B,C,D,E;  reg [4:0] Q;  initial begin  A = 1'b1;  B = 1'b0;  C = 1'b0;  D = 1'b0;  out = 1'b0;  Q=1'b00000;  end      always @(posedge clk or posedge reset)  begin  if(reset)  begin  A <= 1'b1;  B <= 1'b0;  C <= 1'b0;  D <= 1'b0;  out <= 1'b0;  end  else if(A==1'b1)  begin  if(in)  begin  B <= 1'b1;  A <= 1'b0;  out<=1'b0;  end  else  begin  A <= 1'b1;  out <= 1'b0;  end  end  else if(B==1'b1)  begin  if(in)  begin  B<=1'b1;  out<=1'b0;  end  else  begin  B<=1'b0;  C<=1'b1;  out<=1'b0;  end  end  else if(C==1'b1)  begin  if(in)  begin  D<=1'b1;  C<=1'b0;  out<=1'b0;  end  else  begin  C<=1'b0;  A<=1'b1;  out<=1'b0;  end  end  else if(D==1'b1)  begin  if(in)  begin  D<=1'b0;  B<=1'b1;  out<=1'b0;  end  else  begin  D<=1'b0;  E<=1'b1;  out<=1'b0;  end  end  else if(E==1'b1)  begin  if(in)  begin  A<=1'b1;  B<=1'b0;  out<=1'b1;  end  else  begin  A<=1'b1;  E<=1'b0;  out<=1'b0;  end  end  end      always @(posedge clk) begin  if(reset) begin  Q = 4'b00000;  end  else begin  Q[0] <= Q[1];  Q[1] <= Q[2];  Q[2] <= Q[3];  Q[3] <= Q[4];  Q[4] <= in;  end  end  endmodule | `timescale 1ns / 1ps  module mealy\_tb;  reg clk,reset,in;  wire out;  wire [4:0] Q;  mealy u\_test(  .clk ( clk),  .reset ( reset),  .in ( in),    .out (out),  .Q (Q )    );  initial begin  reset = 1'b1;  in=1'b1;  clk = 1'b1;  end  always reset = #200 ~reset;  always in= #20 $urandom%2;  always clk = #10 ~clk;  initial begin  #1000  $finish;  end  endmodule |

3) Simulation

스크린샷이(가) 표시된 사진

자동 생성된 설명

**3. 10101 overlapping Sequence Detector ( in Moore machine)**

이제 Moore machine을 통해 10101 sequence detector을 제작해보겠다. Moore machine은 input이 들어가면 그 다음 turn에 output이 1이 검출되어야 한다. 그래서 이를 처리하기 위해는 Mealy machine보다 더 많은 state가 마련되어야 한다.

1) state diagram and state table

그림, 스케치, 라인 아트, 클립아트이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
| Q(present) | Q(next) | | output |
| X=0 | X=1 |
| A | A | B | 0 |
| B | C | B | 0 |
| C | A | D | 0 |
| D | E | B | 0 |
| E | A | F | 0 |
| F | E | B | 1 |

2) Verilog source

|  |  |
| --- | --- |
| Design code | Testbench code |
| `timescale 1ns / 1ps  module moore(  input clk, reset,  input in,  output out,  output [4:0] Q  );    reg out;  reg A,B,C,D,E,F;  reg [4:0] Q;  initial begin  A = 1'b1;  B = 1'b0;  C = 1'b0;  D = 1'b0;  E = 1'b0;  F = 1'b0;  out = 1'b0;  Q=1'b00000;  end      always @(posedge clk or posedge reset)  begin  if(reset)  begin  A <= 1'b1;  B <= 1'b0;  C <= 1'b0;  D <= 1'b0;    out <= 1'b0;  end  else if(A==1'b1)  begin  if(in)  begin  B <= 1'b1;  A <= 1'b0;  end  else  begin  A <= 1'b1;  end  out <= 1'b0;  end  else if(B==1'b1)  begin  if(in)  begin  B<=1'b1;  end  else  begin  B<=1'b0;  C<=1'b1;  end  out<=1'b0;  end  else if(C==1'b1)  begin  if(in)  begin  D<=1'b1;  C<=1'b0;  end  else  begin  C<=1'b0;  A<=1'b1;  end  out<=1'b0;  end  else if(D==1'b1)  begin  if(in)  begin  D<=1'b0;  B<=1'b1;  end  else  begin  D<=1'b0;  E<=1'b1;  end  out<=1'b0;  end  else if(E==1'b1)  begin  if(in)  begin  F<=1'b1;  E<=1'b0;  end  else  begin  A<=1'b1;  E<=1'b0;  end  out<=1'b0;  end  else if(F==1'b1)  begin  if(in)  begin  B<=1'b1;  F<=1'b0;  end  else  begin  E<=1'b1;  F<=1'b0;  end  out <= 1'b1;  end  end      always @(posedge clk) begin  if(reset) begin  Q = 4'b00000;  end  else begin  Q[0] <= Q[1];  Q[1] <= Q[2];  Q[2] <= Q[3];  Q[3] <= Q[4];  Q[4] <= in;  end  end  endmodule | `timescale 1ns / 1ps  module moore\_tb;  reg clk,reset,in;  wire out;  wire [4:0] Q;  moore u\_test(  .clk ( clk),  .reset ( reset),  .in ( in),    .out (out),  .Q (Q )    );  initial begin  reset = 1'b1;  in=1'b1;  clk = 1'b1;  end  always reset = #200 ~reset;  always in= #20 $urandom%2;  always clk = #10 ~clk;  initial begin  #1000  $finish;  end  endmodule |

3) Simulation

스크린샷이(가) 표시된 사진

자동 생성된 설명