Stopwatch

논리회로실습

부경대 컴퓨터·인공지능공학부 최필주

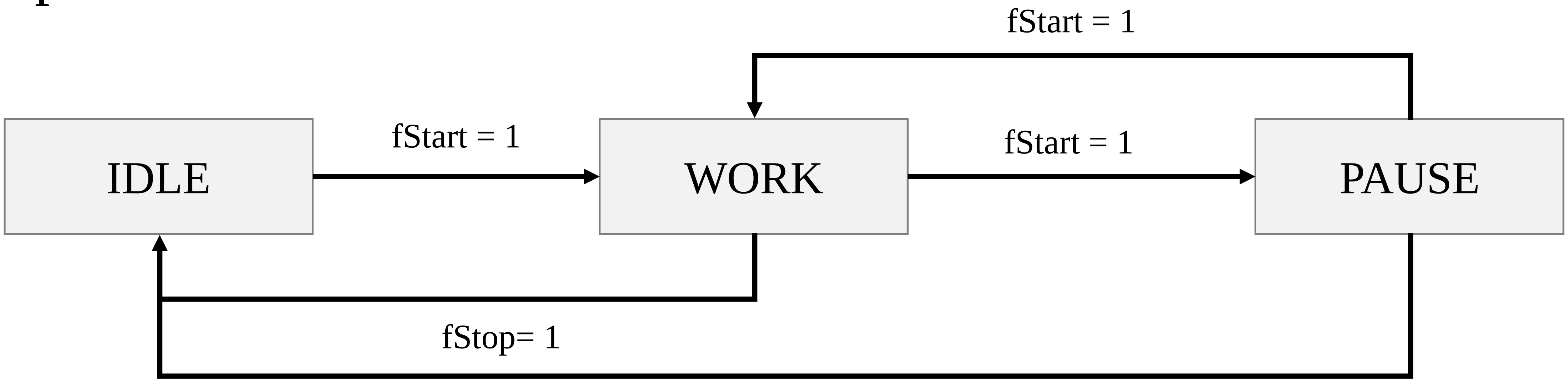
개요

- 구현하고자 하는 기능
  - 동작: 00.0 ~ 99.9까지 셀 수 있는 stopwatch
  - 입력
    - Start: 동작 시작/정지
    - Stop: 중지
  - 출력
    - FND × 3: 00.0 ~ 99.9 초 표현

➔ 동작을 어떻게 나타낼 수 있을까?

Finite State Machine (FSM)

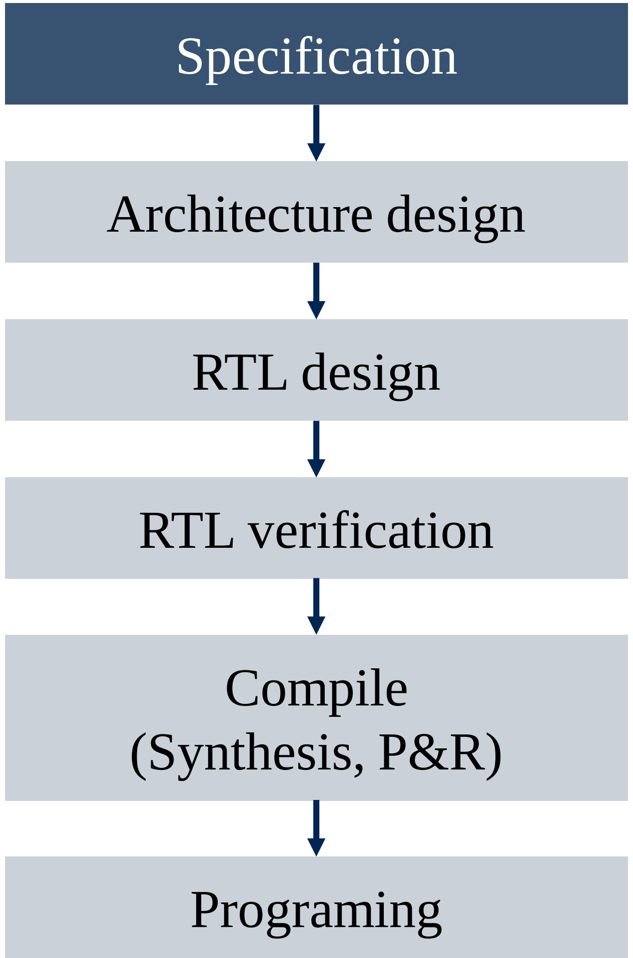
- Stop watch의 FSM



↳ 다양한 상태

Stopwatch의 구현

- 명세
  - 입력
    - i_Clk
    - i_Rst ➔ push[3]
    - i_fStart ➔ push[0]
    - i_fStop ➔ push[1]
  - 출력
    - o_Sec0 ➔ HEX0[6:0]
    - o_Sec1 ➔ HEX1[6:0]
    - o_Sec2 ➔ HEX2[6:0]



Specification

Architecture design

RTL design

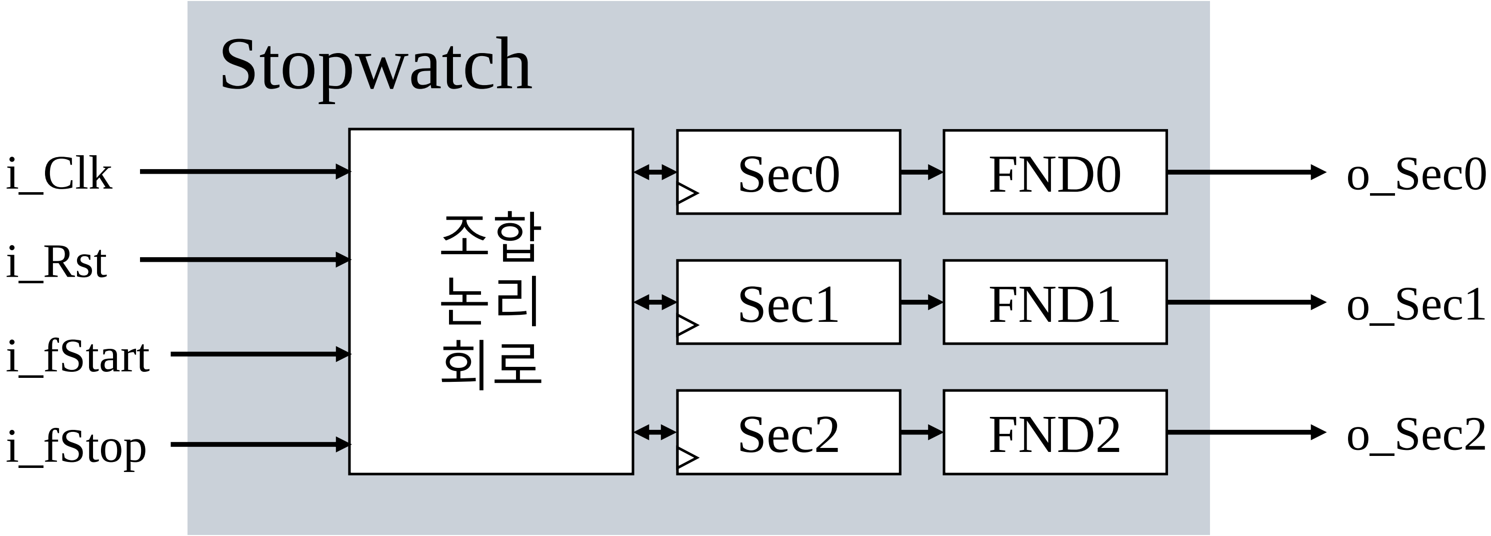
RTL verification

Compile
(Synthesis, P&R)

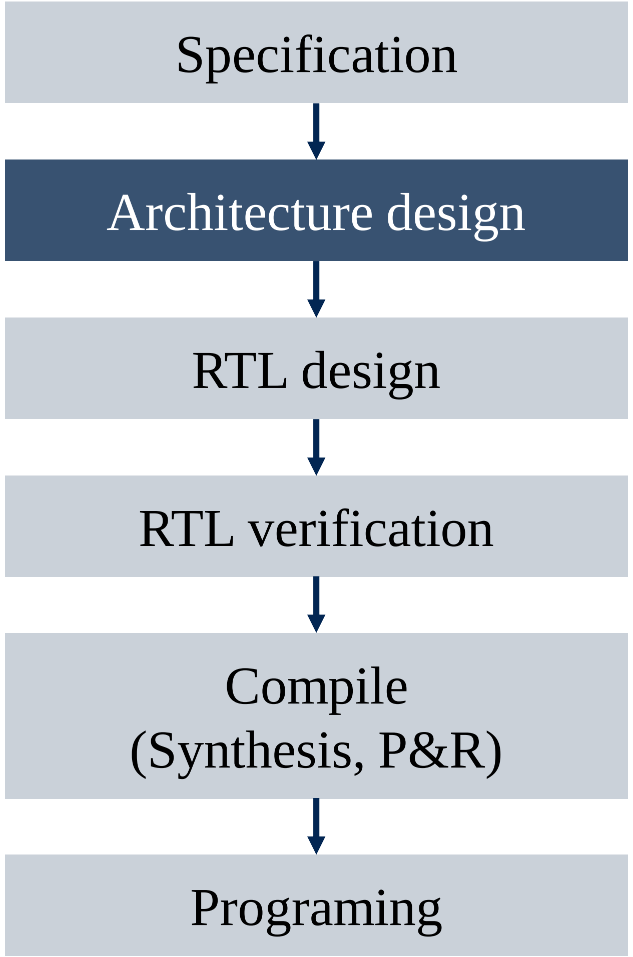
Programing

Stopwatch의 구현

구조 설계



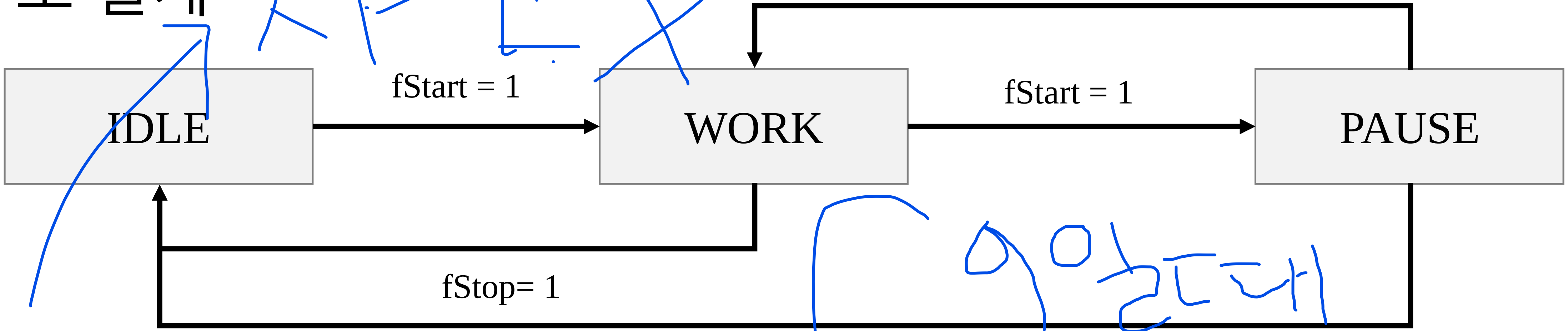
- 필요한 레지스터
  - State
  - ClkCnt
  - Sec0, Sec1, Sec2



Stopwatch의 구현 상태구분 FSM

- 구조 설계

시간 X



Regs.	bits	IDLE	WORK	PAUSE
n_ClkCnt	23	0	fLstClk ? 0 : c_ClkCnt + 1	c_ClkCnt
n_Sec0	4	0	fIncSec0 ? fLstSec0 ? 0 : c_Sec0 + 1 : c_Sec0	c_Sec0
n_Sec1	4	0	fIncSec1 ? fLstSec1 ? 0 : c_Sec1 + 1 : c_Sec1	c_Sec1
n_Sec2	4	0	fIncSec2 ? fLstSec2 ? 0 : c_Sec2 + 1 : c_Sec2	c_Sec2

- fLstClk = c_ClkCnt == 100_000_000/20 – 1
- fLstSec0 = c_Sec0 == 9
- fLstSec1 = c_Sec1 == 9
- fLstSec2 = c_Sec2 == 9

- fIncSec0 = fLstClk
- fIncSec1 = fIncSec0 && fLstSec0
- fIncSec2 = fIncSec1 && fLstSec1

Stopwatch의 구현

- RTL 설계
  - StopWatch.v

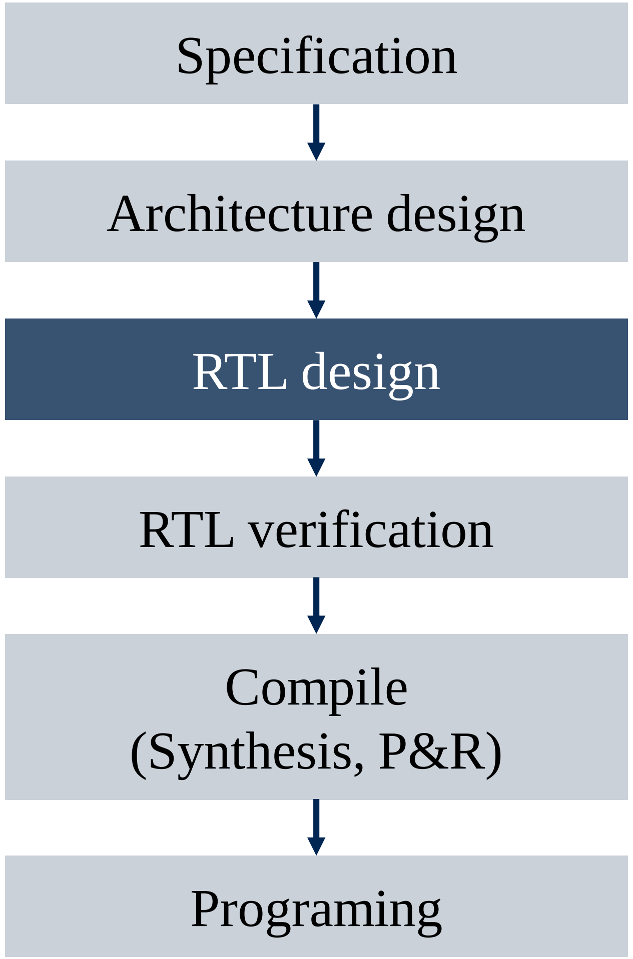
```
module StopWatch(i_Clk, i_Rst, i_fStart, i_fStop, o_Sec0, o_Sec1, o_Sec2);

input          i_Clk, i_Rst;
input          i_fStart, i_fStop;
output wire    [6:0]      o_Sec0, o_Sec1, o_Sec2;

reg            [1:0]      c_State,    n_State;
reg            [3:0]      c_Sec0,     n_Sec0;
reg            [3:0]      c_Sec1,     n_Sec1;
reg            [3:0]      c_Sec2,     n_Sec2;
reg            [22:0]     c_ClkCnt,   n_ClkCnt;
reg                       c_fStart,   n_fStart;
reg                       c_fStop,    n_fStop;

wire           fLstClk;
wire           fLstSec0, fLstSec1, fLstSec2;
wire           fIncSec0, fIncSec1, fIncSec2;
wire           fStart, fStop;

parameter      LST_CLK    = 100_000_000/20 - 1;
parameter      IDLE = 2'b00, WORK = 2'b01, PAUSE = 2'b10;
```



Stopwatch의 구현

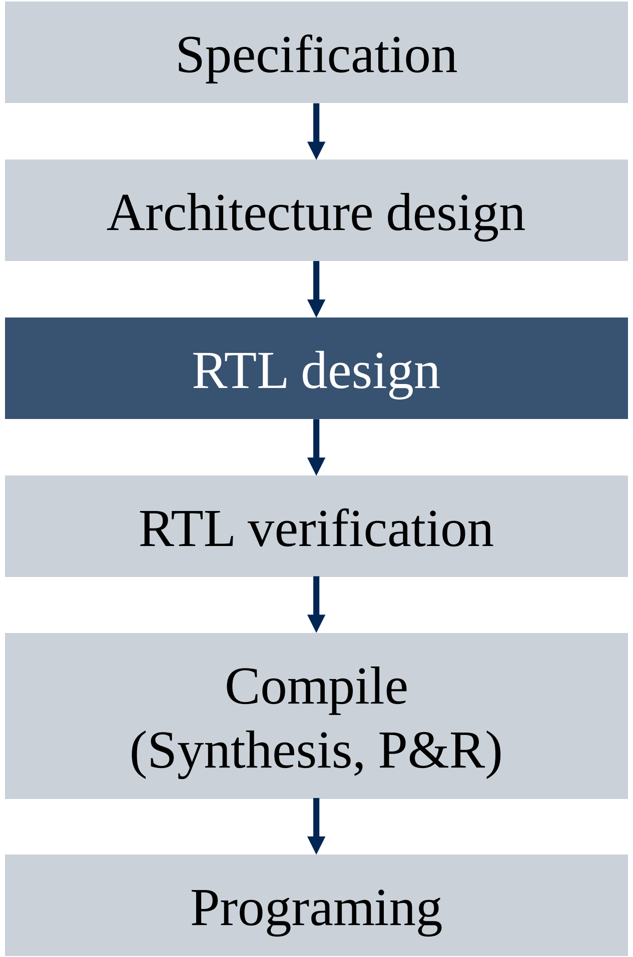
- RTL 설계
  - StopWatch.v

```
FND         FND0(c_Sec0,            o_Sec0);
FND         FND1(c_Sec1,            o_Sec1);
FND         FND2(c_Sec2,            o_Sec2);

always@(posedge i_Clk, negedge i_Rst)
            if(!i_Rst) begin
                        c_State     = IDLE;
                        c_ClkCnt    = 0;
                        c_Sec0      = 0;
                        c_Sec1      = 0;
                        c_Sec2      = 0;
                        c_fStart    = 1;
                        c_fStop     = 1;
            end else begin
                        c_State     = n_State   ;
                        c_ClkCnt    = n_ClkCnt  ;
                        c_Sec0      = n_Sec0    ;
                        c_Sec1      = n_Sec1    ;
                        c_Sec2      = n_Sec2    ;
                        c_fStart    = n_fStart  ;
                        c_fStop     = n_fStop   ;
            end
```

선언

reg 생성



Specification

Architecture design

RTL design

RTL verification

Compile
(Synthesis, P&R)

Programing

Stopwatch의 구현

RTL 설계

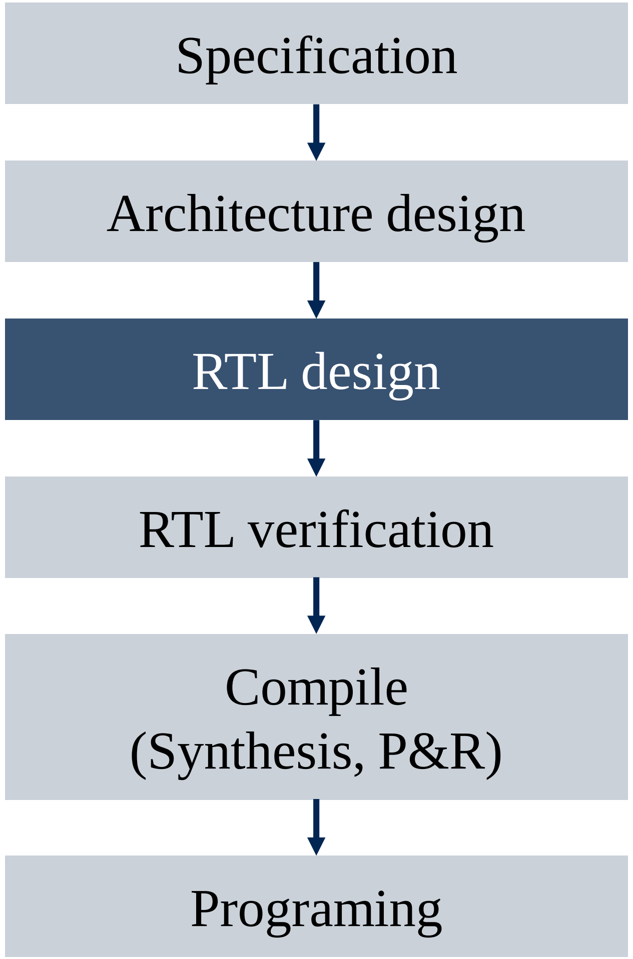
- StopWatch.v

```
assign      fStart       = !i_fStart && c_fStart,
            fStop        = !i_fStop  && c_fStop;
assign      fLstClk      = c_ClkCnt == LST_CLK,
            fLstSec0= c_Sec0 == 9,
            fLstSec1= c_Sec1 == 9,
            fLstSec2= c_Sec2 == 9;
assign      fIncSec0= fLstClk,
            fIncSec1= fIncSec0 && fLstSec0,
            fIncSec2= fIncSec1 && fLstSec1;

always@*
begin
            n_fStart     = i_fStart    ;
            n_fStop      = i_fStop     ;

            n_State      = c_State     ;
            n_ClkCnt     = c_ClkCnt ;
            n_Sec0       = c_Sec0      ;
            n_Sec1       = c_Sec1      ;
            n_Sec2       = c_Sec2      ;
```

중복되는
코드가
있는경우



Specification

Architecture design

RTL design

RTL verification

Compile
(Synthesis, P&R)

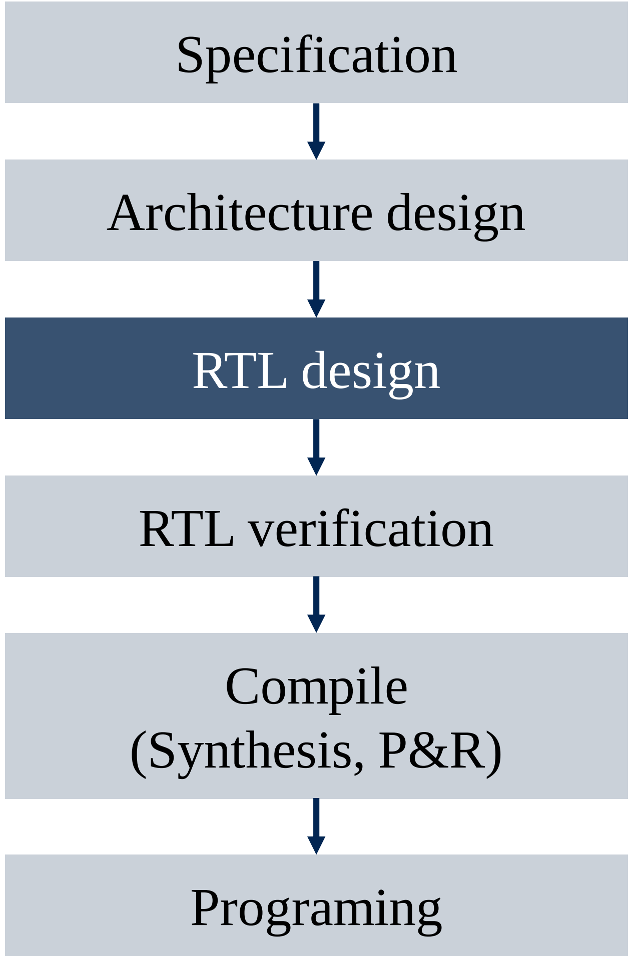
Programing

Stopwatch의 구현

- RTL 설계
  - StopWatch.v

```
        case(c_State)
                IDLE: begin
                        n_ClkCnt    = 0;
                        n_Sec0      = 0;
                        n_Sec1      = 0;
                        n_Sec2      = 0;
                        if(fStart)  n_State = WORK;
                end
                WORK: begin
                        n_ClkCnt    = fLstClk ? 0 : c_ClkCnt + 1;
                        n_Sec0      = fIncSec0 ? fLstSec0 ? 0 : c_Sec0 + 1 : c_Sec0;
                        n_Sec1      = fIncSec1 ? fLstSec1 ? 0 : c_Sec1 + 1 : c_Sec1;
                        n_Sec2      = fIncSec2 ? fLstSec2 ? 0 : c_Sec2 + 1 : c_Sec2;

                        if(fStop)   n_State = IDLE;
                        else if(fStart)n_State = PAUSE;
                end
                PAUSE:
                        if(fStop)   n_State = IDLE;
                        else if(fStart)n_State = WORK;
        endcase
end
endmodule
```



Specification

Architecture design

RTL design

RTL verification

Compile
(Synthesis, P&R)

Programing

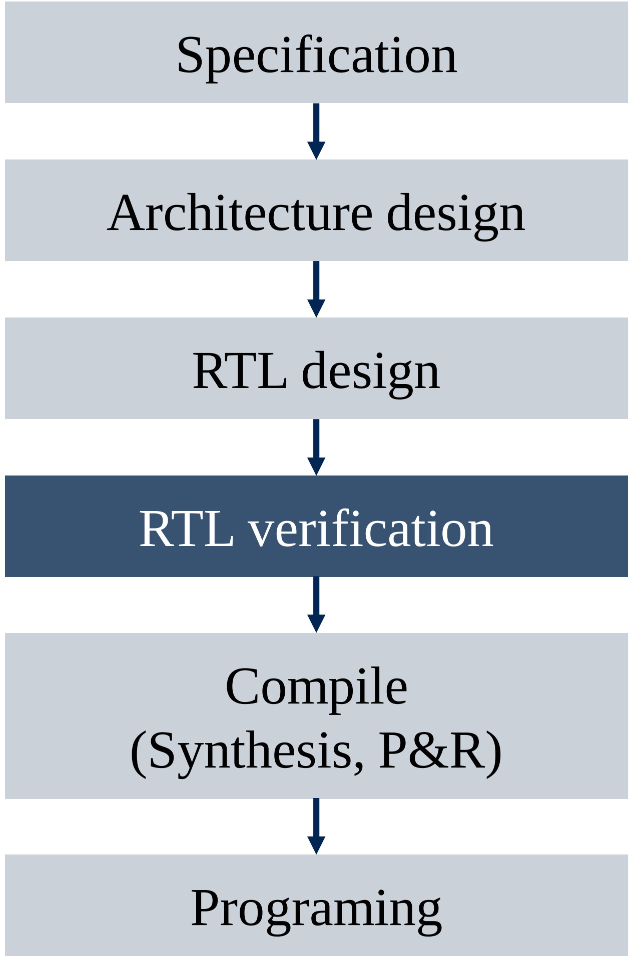
Stopwatch의 구현

- RTL 검증 - testbench module
  - tb_StopWatch

```
`timescale 1ns / 1ns
module tb_StopWatch;
reg                     Clk;
reg                     Rst;
reg                     fStart;
reg                     fStop;

StopWatch U0(Clk, Rst, fStart, fStop, , , );
always #10 Clk = ~Clk;
initial
begin
          Clk         = 1;
          Rst         = 0;
          fStart      = 1;
          fStop       = 1;

          @(negedge Clk) Rst = 1;
          #100                fStart      = 1;         #20 fStart = 1;
          #222_222_200        fStart      = 0;         #20 fStart = 1;
          #1000               fStart      = 0;         #20 fStart = 1;
          #333_333_300        fStop       = 0;         #20 fStop = 1;
          $stop;
end
endmodule
```


Specification

Architecture design

RTL design

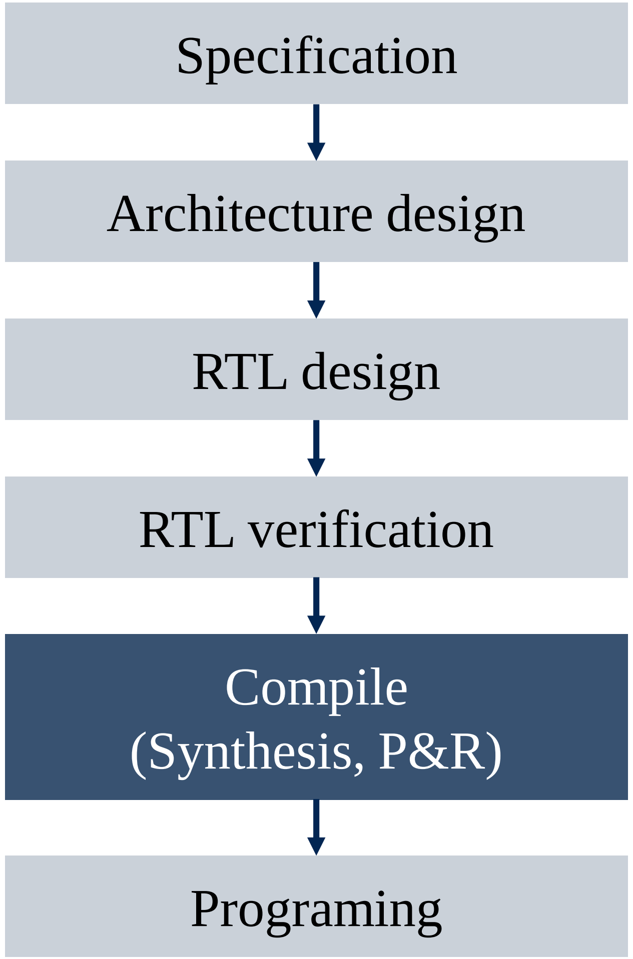
RTL verification

Compile
(Synthesis, P&R)

Programing

Stopwatch의 구현

- FPGA 구현 – pin 설정
  - 입력: i_Clk, i_Rst, i_fStart, i_fStop
    - i_fStart, i_fStop, i_Rst은 push 버튼 3개 사용
  - 출력: o_FND(7x3)



Specification

Architecture design

RTL design

RTL verification

Compile
(Synthesis, P&R)

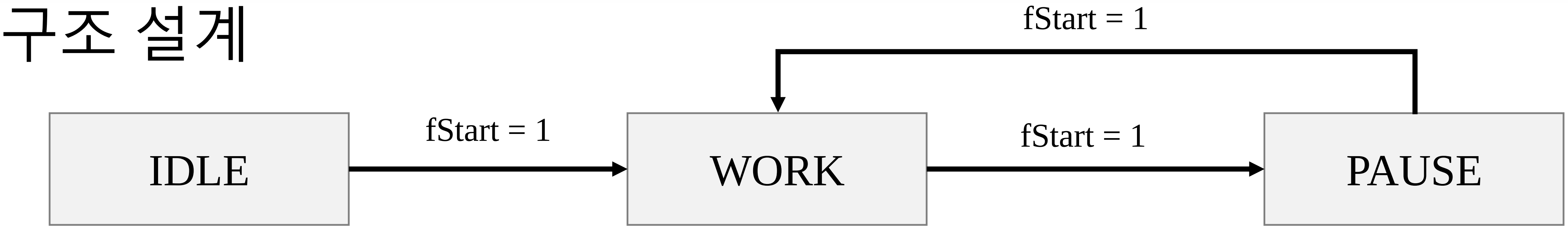
Programing

추가 연습

- 변경 사항
  - 입력: i_fStop ➔ i_fRecord
  - 출력: o_Sec0, o_Sec1, o_Sec2에 o_Sec3, o_Sec4, o_Sec5 추가
  - 중지조건 변경: 정지된 상태에서 fRecord 누르기
  - 추가 동작: 동작 중 fRecord를 누르면 그 때 시각 정보를 o_Sec3, o_Sec4, o_Sec5에 표시

Stopwatch의 구현

- 구조 설계



fStart = 1

IDLE

fStart = 1

WORK

fStart = 1

PAUSE

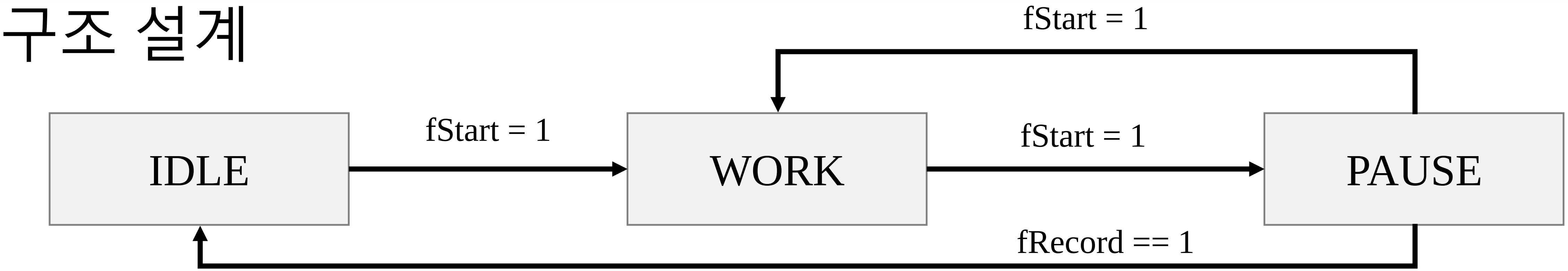
Regs.	bits	IDLE	WORK	PAUSE
n_ClkCnt	23	0	fLstClk ? 0 : c_ClkCnt + 1	c_ClkCnt
n_Sec0	4	0	fIncSec0 ? fLstSec0 ? 0 : c_Sec0 + 1 : c_Sec0	c_Sec0
n_Sec1	4	0	fIncSec1 ? fLstSec1 ? 0 : c_Sec1 + 1 : c_Sec1	c_Sec1
n_Sec2	4	0	fIncSec2 ? fLstSec2 ? 0 : c_Sec2 + 1 : c_Sec2	c_Sec2

- fLstClk = c_ClkCnt == 100_000_000/20 – 1
- fLstSec0 = c_Sec0 == 9
- fLstSec1 = c_Sec1 == 9
- fLstSec2 = c_Sec2 == 9

- fIncSec0 = fLstClk
- fIncSec1 = fIncSec0 && fLstSec0
- fIncSec2 = fIncSec1 && fLstSec1

Stopwatch의 구현

- 구조 설계



Regs.	bits	IDLE	WORK	PAUSE
n_ClkCnt	23	0	fLstClk ? 0 : c_ClkCnt + 1	c_ClkCnt
n_Sec0	4	0	fIncSec0 ? fLstSec0 ? 0 : c_Sec0 + 1 : c_Sec0	c_Sec0
n_Sec1	4	0	fIncSec1 ? fLstSec1 ? 0 : c_Sec1 + 1 : c_Sec1	c_Sec1
n_Sec2	4	0	fIncSec2 ? fLstSec2 ? 0 : c_Sec2 + 1 : c_Sec2	c_Sec2
n_Sec3	4	10	fRecord ? c_Sec0 : c_Sec3	c_Sec3

- fLstClk = c_ClkCnt == 100_000_000/20 – 1
- fLstSec0 = c_Sec0 == 9
- fLstSec1 = c_Sec1 == 9
- fLstSec2 = c_Sec2 == 9

- fIncSec0 = fLstClk
- fIncSec1 = fIncSec0 && fLstSec0
- fIncSec2 = fIncSec1 && fLstSec1