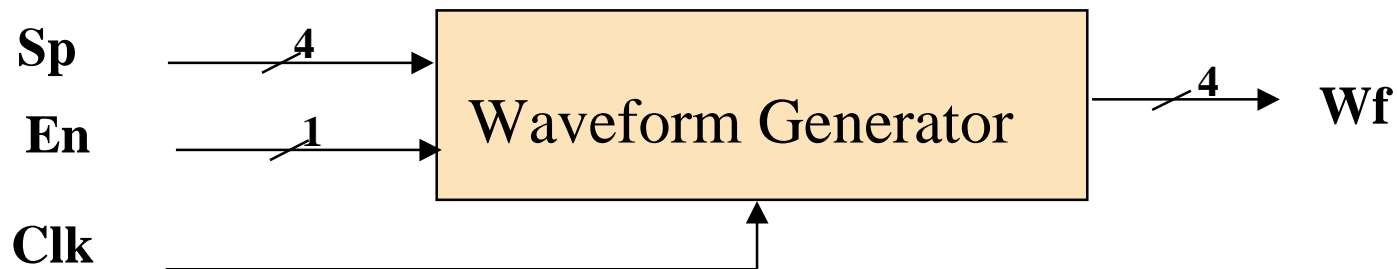

Lecture 17

Top Down Design with behavioral descriptions (by EXAMPLE)

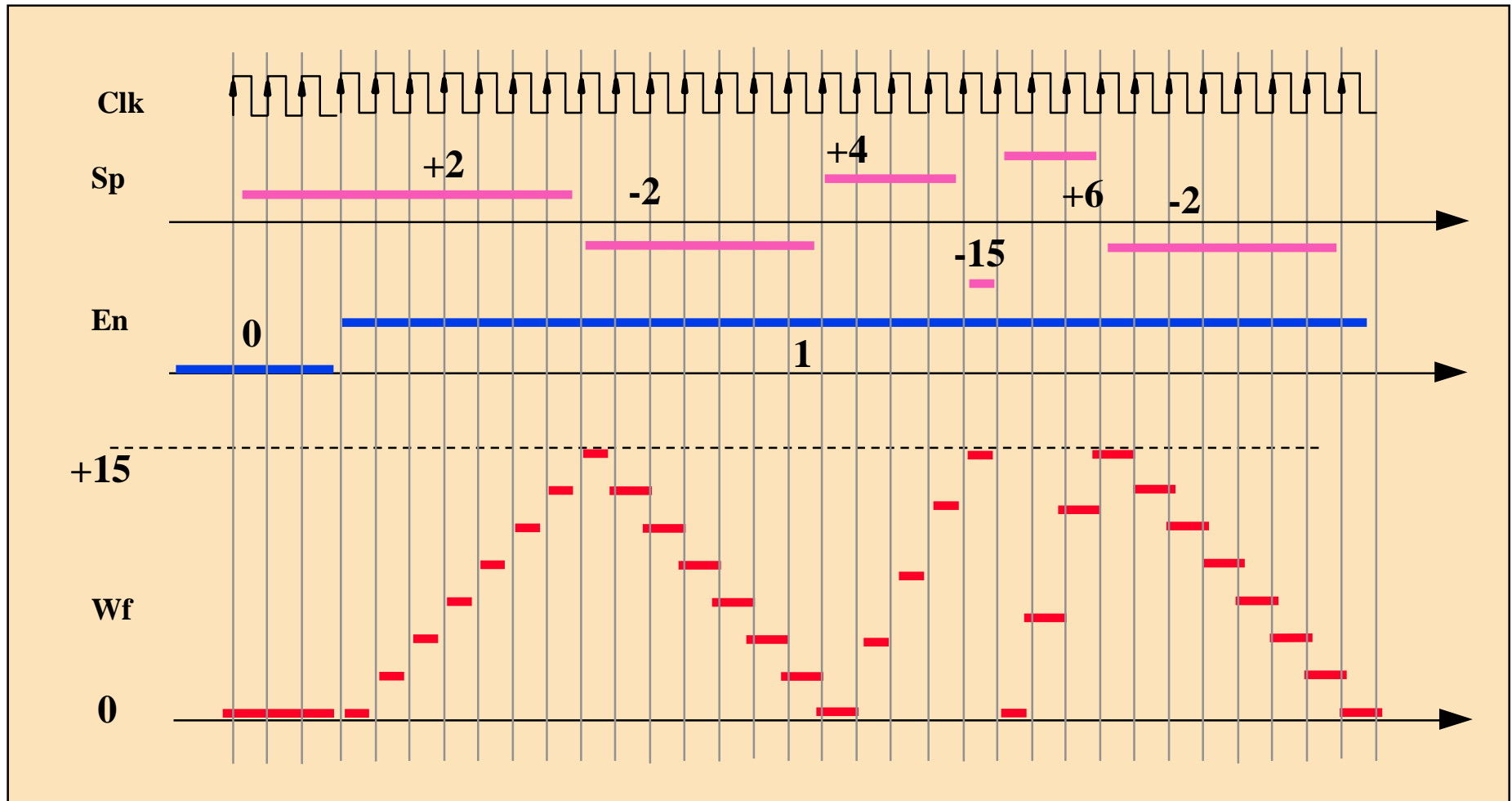
Slides
from Adam
Postula

WAVEFORM GENERATOR

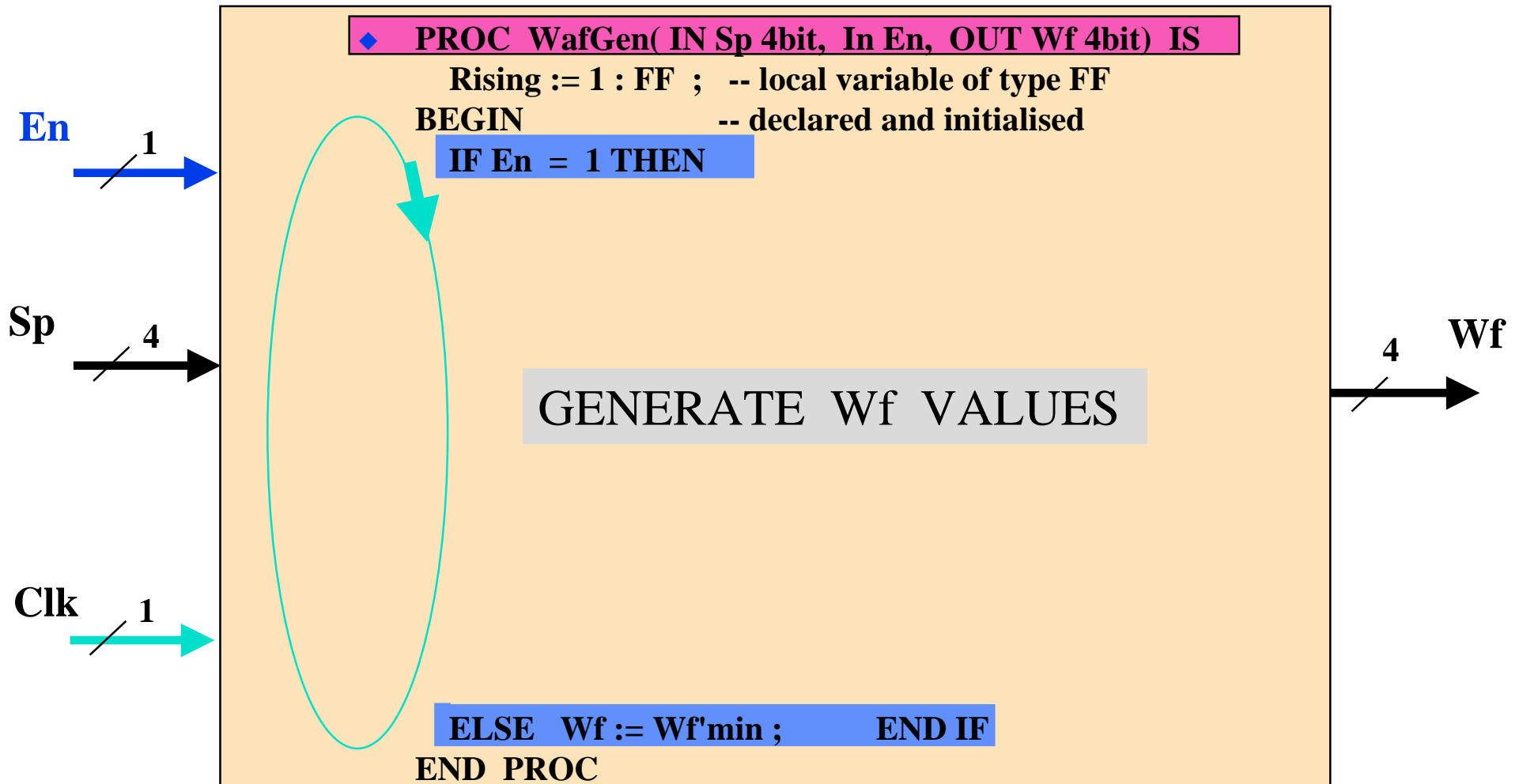
- ◆ Design a waveform generator with a 4 bit data output Wf.
- ◆ a. The generator starts from $Wf = 0$ (0000)
- b. A 4 bit value on the Sp input is added to the previous value of Wf every clock if Wf is smaller than +15 (1111) and larger than 0.
- c. The same Sp value is subtracted when Wf reached or exceeded +15.



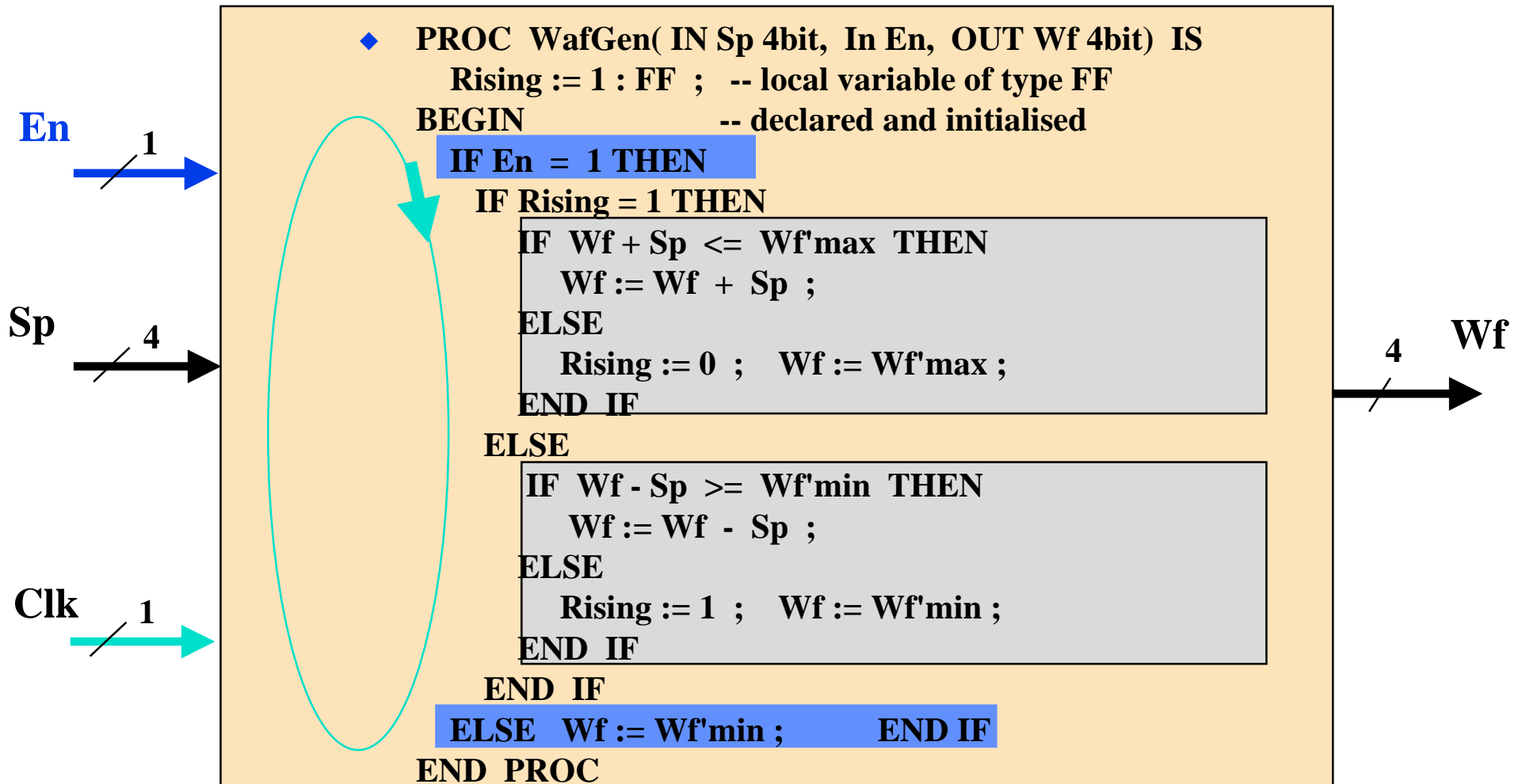
WAVEFORM SAMPLES



WAVEFORM GENERATOR



WAVEFORM GENERATOR



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 0



Sp = 6



Clk



Wf = 0000



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 0000



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 0110



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 0110



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1100



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1100



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1111



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1111



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1001



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 1001



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 0011



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



Clk



Wf = 0011



WAVEFORM GENERATOR

```
◆ PROC WafGen( IN Sp 4bit, In En, OUT Wf 4bit) IS
  Rising := 1 : FF ; -- local variable of type FF
  BEGIN
    IF En = 1 THEN
      IF Rising = 1 THEN
        IF Wf + Sp <= Wf'max THEN
          Wf := Wf + Sp ;
        ELSE
          Rising := 0 ; Wf := Wf'max ;
        END IF
      ELSE
        IF Wf - Sp >= Wf'min THEN
          Wf := Wf - Sp ;
        ELSE
          Rising := 1 ; Wf := Wf'min ;
        END IF
      END IF
    ELSE Wf := Wf'min ; END IF
  END PROC
```

En = 1



Sp = 6



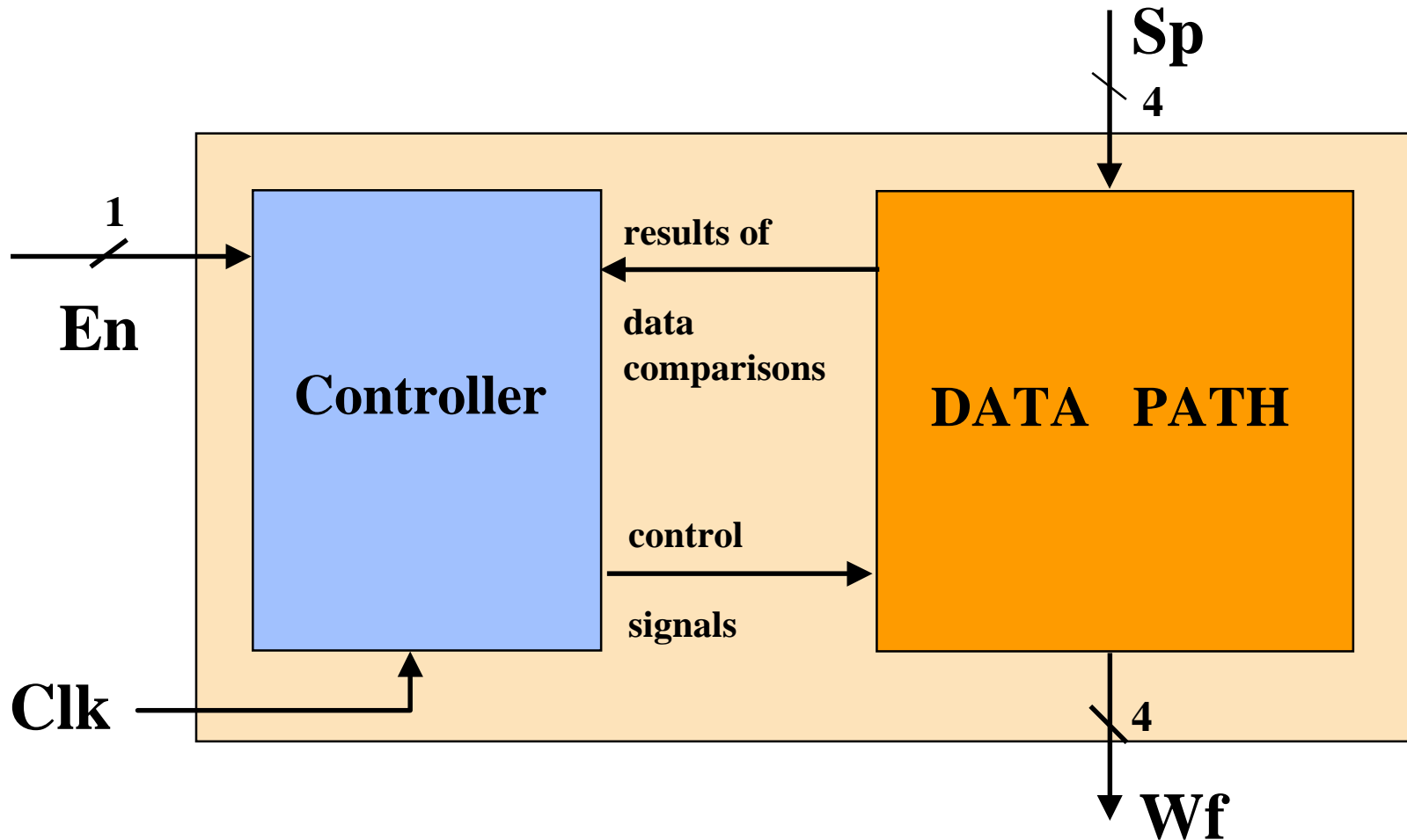
Clk



Wf = 0000



WAVEFORM GENERATOR



DATA PATH

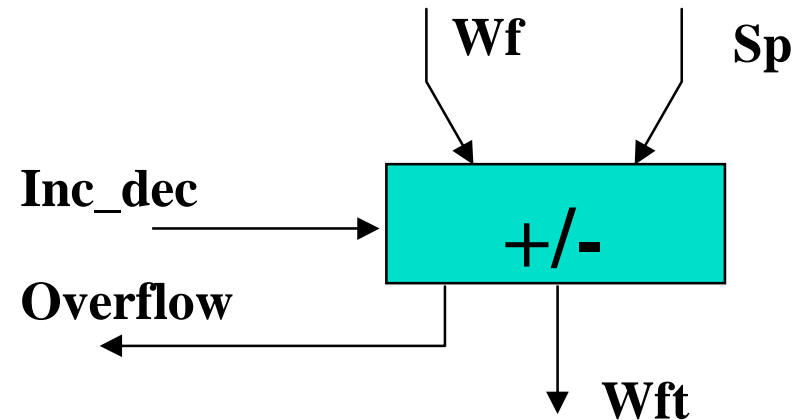
- ◆ The value of Wf is incremented/decremented by the Sp value.
- ◆ Behavioural description :

```
IF Inc_dec = '1' THEN  
    Wft = Wf + Sp  
ELSE  
    Wft = Wf - Sp  
END IF
```

DATA PATH

- ◆ The value of WF is incremented/decremented by the SP value.
- ◆ Behavioural description :

```
IF Inc_dec = '1' THEN
    Wft = Wf + Sp
ELSE
    Wft = Wf - Sp
END IF
```



- ◆ HOW TO PROVIDE THE CALCULATED VALUE OF Wft TO THE WF INPUT OF THE ADDER/SUBTRACTOR ?
- ◆ HOW TO HANDLE OVERFLOW ?

DATA PATH

- ◆ Every clock the WF value is incremented/decremented by the SP value.

- ◆ Behavioural description :

IF Clock'rising THEN

IF NOT Overflow THEN

Latch = '1'

-- WF := Wft

ELSE

-- ? WF = WF' min

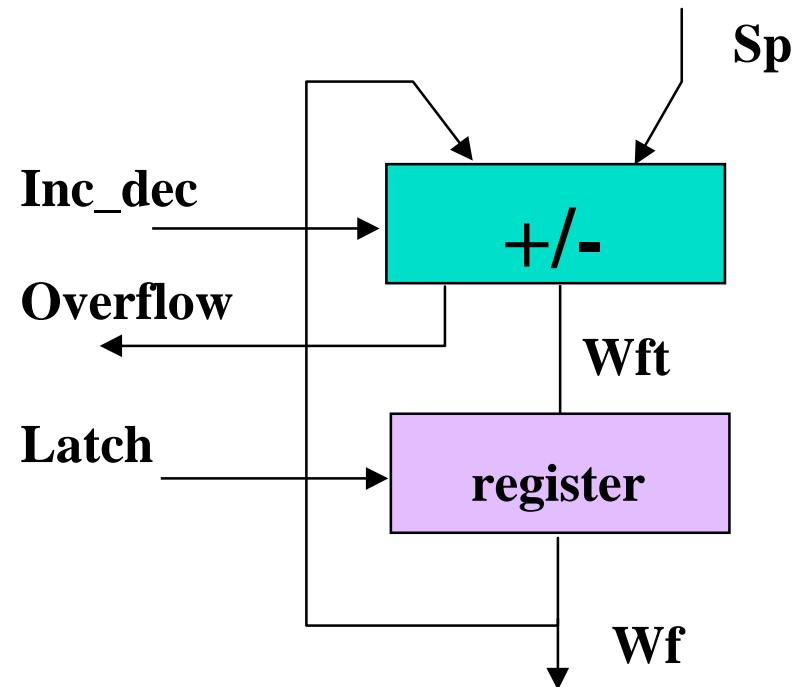
-- ? WF = WF' max

-- ? WF = WF + SP, WF = WF - SP

END IF

END IF

- ◆ HOW TO HANDLE OVERFLOW ?



DATA PATH

- ◆ HOW TO HANDLE OVERFLOW ?

- ◆ Behavioural description :

```
IF overflow THEN
```

```
-- reverse increments
```

```
IF was_decrementing THEN
```

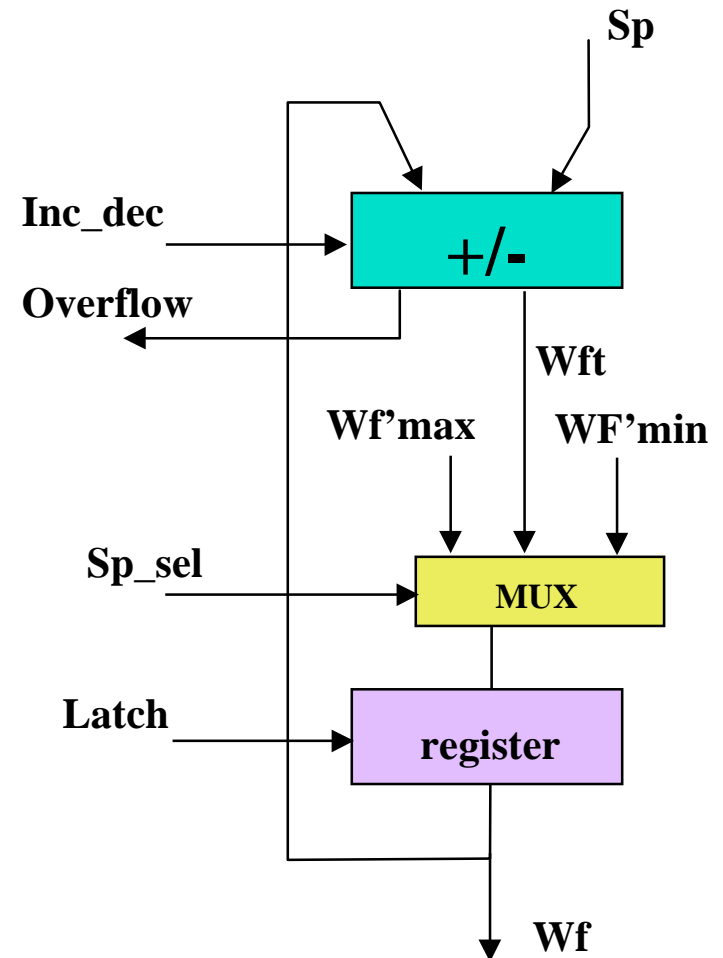
```
-- put maximum WF value on WF
```

```
ELSE
```

```
-- put minium WF value on WF
```

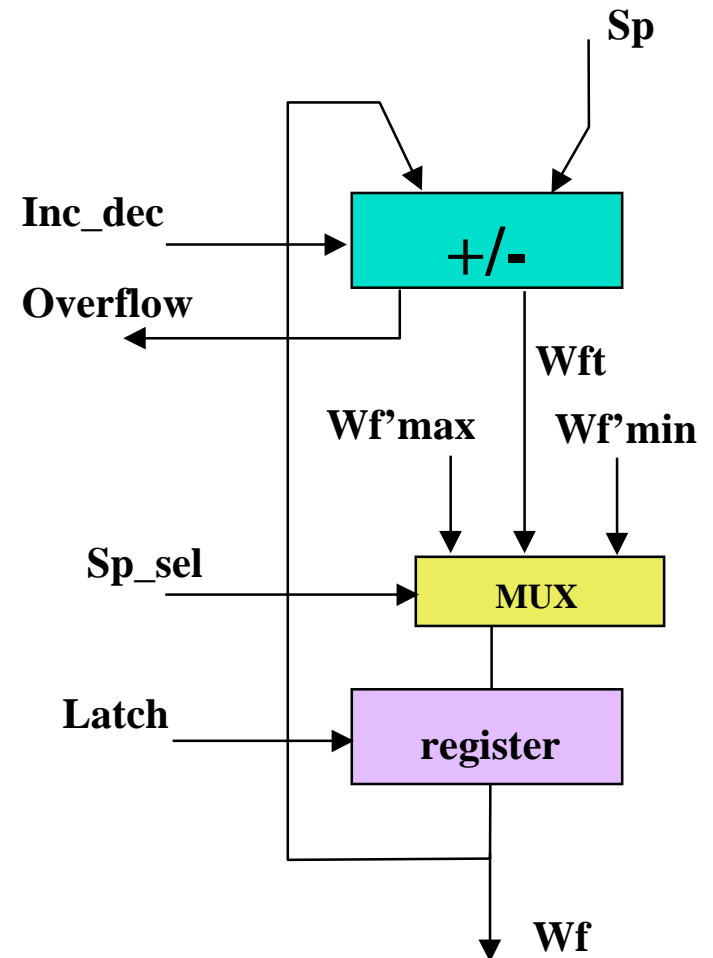
```
END IF
```

```
END IF
```



DATA PATH

- ◆ HOW TO HANDLE OVERFLOW ?
- ◆ Behavioural description :
IF Clock'rising THEN
 IF NOT Overflow THEN -- WF = WF +/- SP
 Latch = '1 ;
 Sp_sel = sel_WFt ;
 ELSE
 Latch = '1 ;
 Inc_dec = NOT Inc_dec
 IF Inc_dec = '1' THEN
 Sp_sel = sel_WF'min
 ELSE
 Sp_sel = sel_WF'max
 END IF
 END IF
END IF



DATA PATH

- ◆ HOW TO HANDLE OVERFLOW ?

- ◆ Behavioural description :

IF Clock'rising **THEN**

IF NOT Overflow **THEN** -- $WF = WF \pm SP$

Latch = '1 ;

Sp_sel = sel_WFt ;

ELSE

Latch = '1 ;

Inc_dec = NOT Inc_dec

IF Inc_dec = '1' **THEN**

Sp_sel = sel_WF'min

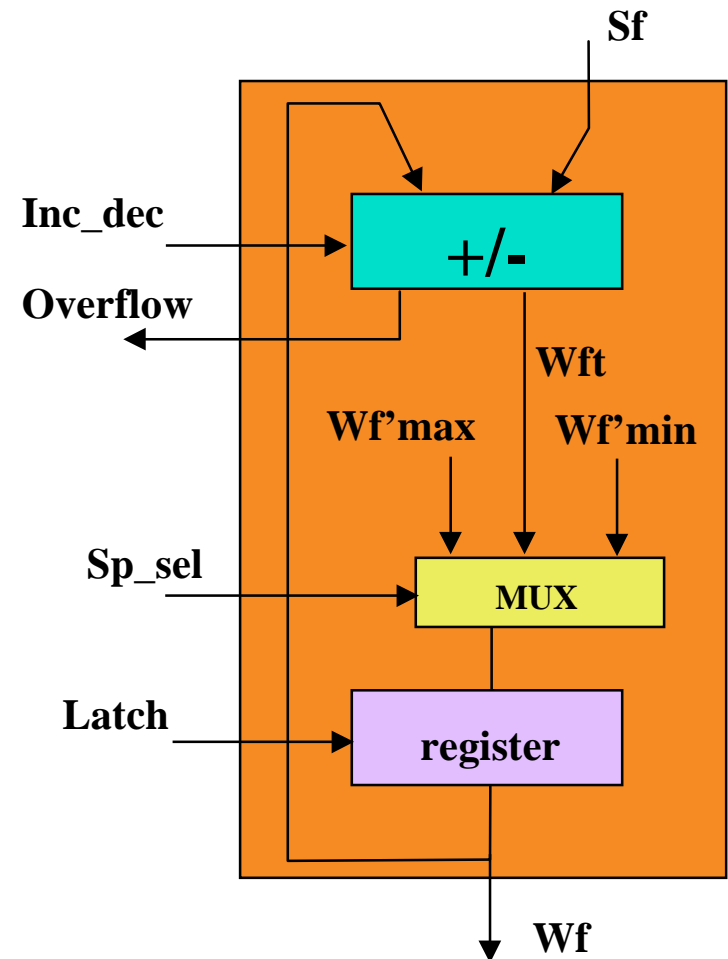
ELSE

Sp_sel = sel_WF'max

END IF

END IF

END IF



CONTROLLER

- ◆ HOW TO HANDLE OVERFLOW ?

- ◆ Behavioural description :

IF Clock'rising THEN

IF NOT Overflow THEN -- WF = WF +/- SP

Latch = '1 ;

Sp_sel = sel_WFt ;

ELSE

Latch = '1 ;

Inc_dec = NOT Inc_dec

IF Inc_dec = '1' THEN

Sp_sel = sel_WF'min

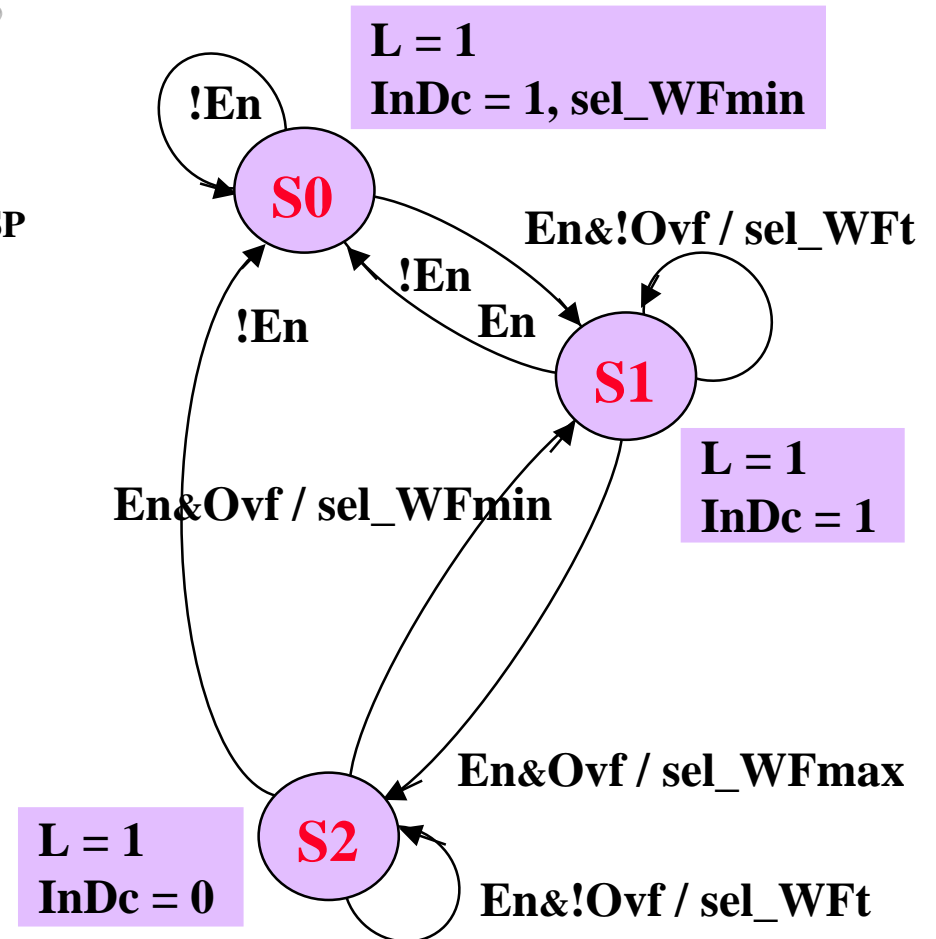
ELSE

Sp_sel = sel_WF'max

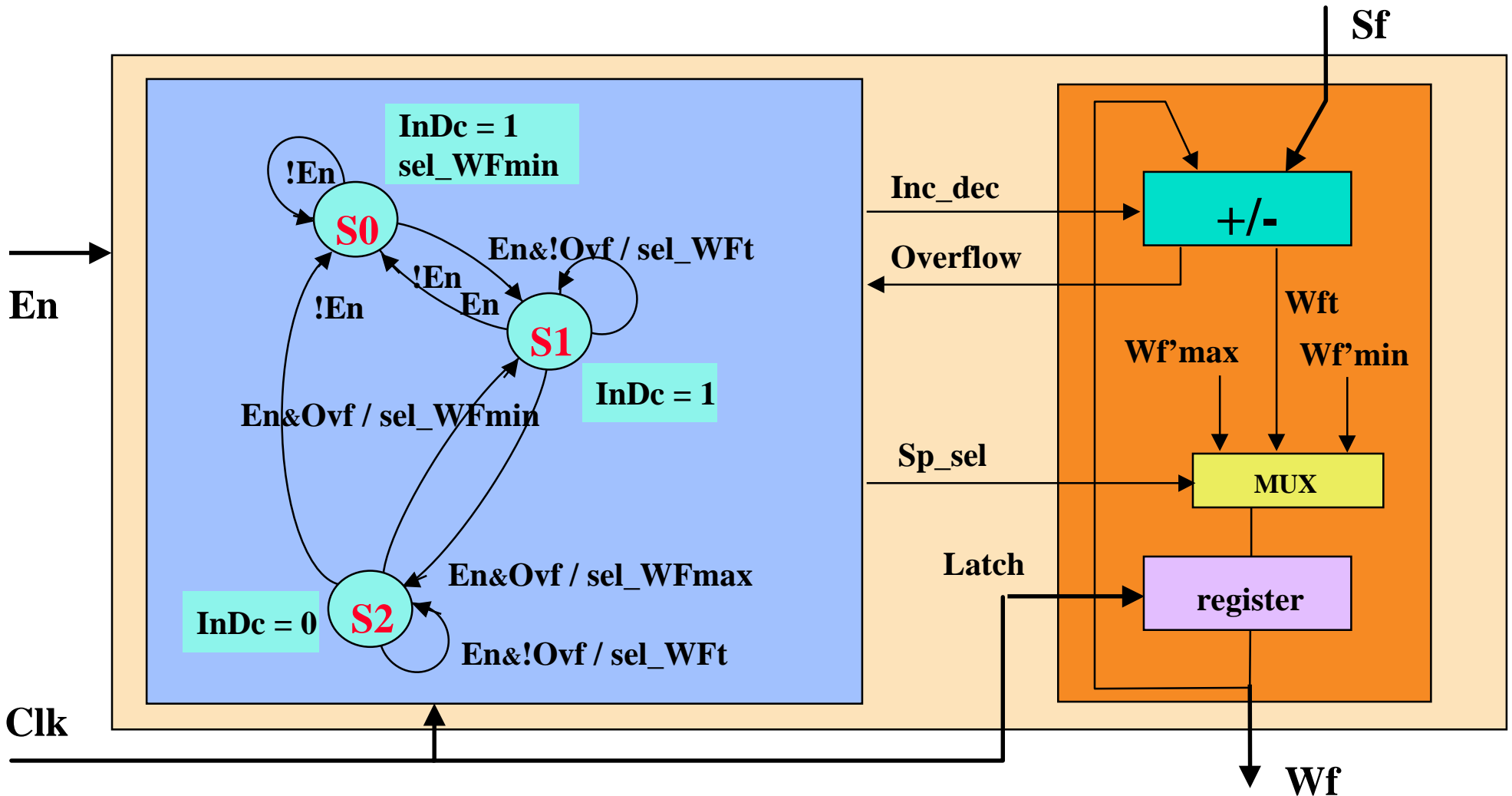
END IF

END IF

END IF



WAVEFORM GENERATOR



What have we learnt?

- ◆ Top Down designs can be made very systematic.
- ◆ The canonical structure Controller-DataPath is a useful model for all sequential designs.
- ◆ Use of behavioral models on different level of abstraction allows to experiment with the functionality to be implemented even before the final design is finished.
- ◆ Use of the "simulate first then design" methodology allows to avoid many mistakes in the early stages of the design process.
- ◆ Proper simulation tools are the necessary prerequisite but they are available on the market and the methodology is widely accepted.