Control Design: Hardwired Approach

Bidyut Kr. Patra

Book: P.Hayes. Computer Architecture and Organization, McGraw-Hill

Control Design: Hardwired Approach

Control Design: Hardwired Approach

- **Method 1:** Classical Method of sequential circuit design. (Optimized in terms of Flip-flop (FF) requirement)
- **Method 2:** One hot method. (Simple circuit)

Classical Method:GCD Processor

- Design a control unit for GCD processor using Classical method.

GCD Processor

Procedure for computing GCD in HDL

gcd(in:X, Y;out:Z)

1. register XR, YR, $TEMPR$;
2. $XR := X$; $YR := Y$; {Input the data}
3. while($XR > 0$) do begin
   if ($XR \leq YR$) then begin
   1. TEMPR:=YR;
   2. YR:=XR;
   3. XR:=TEMPR; { Swap XR and YR}
4. XR:=XR-YR; {Subtract}

$Z := YR$ {Output the result}

end gcd;

Example

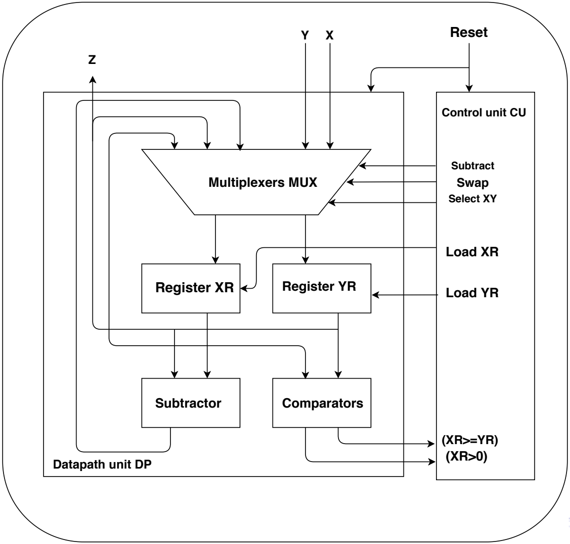
$X = 20, Y = 12$

Conditions		Actions	
		$XR := 20; YR = 12;$	
$XR > 0:$	$XR > YR:$	$XR := XR - YR = 8;$	
$XR > 0:$	$XR <= YR:$	$XR := 12; YR = 8;$	$;XR = XR - YR = 4;$
$XR > 0:$	$XR <= YR:$	$XR := 8; YR = 4;$	$;XR = XR - YR = 4;$
$XR > 0:$	$XR <= YR:$	$XR := 4; YR = 4;$	$;XR = XR - YR = 0;$
$XR <= 0$		$Z = 4$	

Table: Example

$\underline{GCD(20, 12) = 4.}$

Hardware for GCD Processor



State Table for Control Unit of GCD Processor

Design the State Table defining the Control Unit of the GCD Processor.

GCD Processor

Procedure for computing GCD in HDL

gcd(in:X, Y;out:Z)

1. register XR, YR, $TEMPR$;
2. $XR := X$; $YR := Y$; {Input the data: S_0}
3. while($XR > 0$) do begin
   if ($XR \leq YR$) then begin
   1. TEMPR:=YR;
   2. YR:=XR;
   3. XR:=TEMPR; { Swap XR and YR: S_1}
4. XR:=XR-YR; {Subtract: S_2}

$Z := YR$ {Output the result:S_3}

end gcd;

State Table

Table: State Table of Control Unit (GCD Processor)

State	Inputs $XR > 0$ $XR \geq YR$			Outputs				
	0-	10	11	Subtract	Swap	SelectXY	LoadXR	LoadYR
S_0	S_3	S_1	S_2	0	0	1	1	1
S_1	S_2	S_2	S_2	0	1	0	1	1
S_2	S_3	S_1	S_2	1	0	0	1	0
S_3	S_3	S_3	S_3	0	0	0	0	0

Steps of Classical Design Method

1. Construct a P-row state table that defines the desired input-output behaviour.
2. Select minimum number p of D-type flip-flops and assign p-bit binary code to each state. $\{S_0 : 00, S_1 : 01, S_2 : 10, S_3 : 11\}$
3. Design a combinational circuit C that generate the primary output signal $\{z_i\}$ and seconadary outputs $\{D_i\}$ that must be applied to the FFs.

Table: Excitation Table for the control unit of GCD Processor

Inputs		PS		NS		Outputs				
$XR > 0$	$(XR >= YR)$	D_1	D_0	D_1^+	D_0^+	Sub	Sw	XY	XR	YR
0	d	0	0	1	1	0	0	1	1	1
0	d	0	1	1	0	0	1	0	1	1
0	d	1	0	1	1	1	0	0	1	0
0	d	1	1	1	1	0	0	0	0	0
1	0	0	0	0	1	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1
1	0	1	0	0	1	1	0	0	1	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	1	1	1
1	1	0	1	1	0	0	1	0	1	1
1	1	1	0	1	0	1	0	0	1	0
1	1	1	1	1	1	0	0	0	0	0

Steps of Classical Design Method

1. Construct a P-row state table that defines the desired input-output behaviour. ✓
2. Select minimum number p of D-type flip-flops and assign p-bit binary code to each state. ✓
3. Design a combinational circuit C that generate the primary output signal $\{z_i\}$ and seconadary outputs $\{D_i\}$ that must be applied to the FFs.

Table: Excitation Table for the control unit of GCD Processor

Inputs		PS		NS		Outputs				
$XR > 0$	$(XR >= YR)$	D_1	D_0	D_1^+	D_0^+	Sub	Sw	XY	XR	YR
0	d	0	0	1	1	0	0	1	1	1
0	d	0	1	1	0	0	1	0	1	1
0	d	1	0	1	1	1	0	0	1	0
0	d	1	1	1	1	0	0	0	0	0
1	0	0	0	0	1	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1
1	0	1	0	0	1	1	0	0	1	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	1	1	1
1	1	0	1	1	0	0	1	0	1	1
1	1	1	0	1	0	1	0	0	1	0
1	1	1	1	1	1	0	0	0	0	0

Step 3: Generate output signals

$$D_1^+ = \overline{XR > 0} + (XR \geq YR) + D_0$$

$$D_0^+ = D_1.D_0 + \overline{(XR \geq XY)}.\overline{D_0} + \overline{(XR > 0)}.\bar{D}_0$$

$$Subtract = D_1.\bar{D}_0$$
$$Swap = \bar{D}_1.D_0$$
$$SelectXY = \bar{D}_1.\bar{D}_0$$
$$LoadXR = \bar{D}_0 + \bar{D}_1$$
$$LoadYR = \bar{D}_1$$

Table: Outputs of the Control ckt.

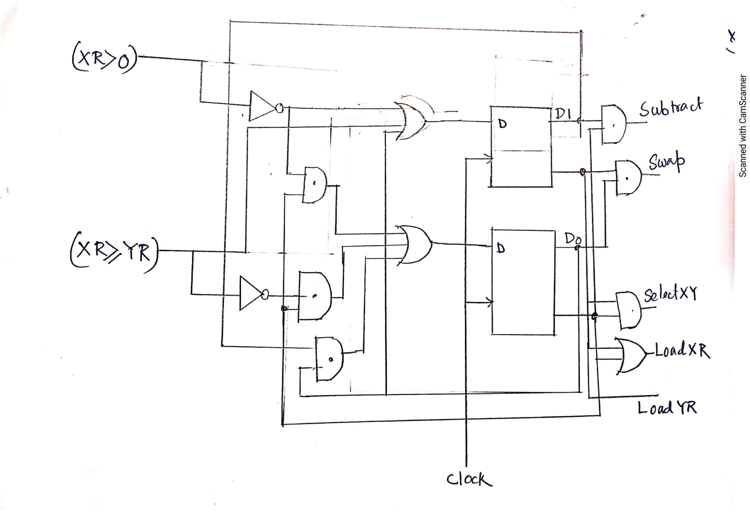

(XR>0)
(XR≥YR)
D
D1
D
D0
Subtract
Swap
SelectXY
LoadXR
LoadYR
Clock
Scanned with CamScanner

Figure:

One-hot Method

THANK YOU