Design of Control Path

Debdeep Mukhopadhyay

Hardwired Hardware

GCD Processor

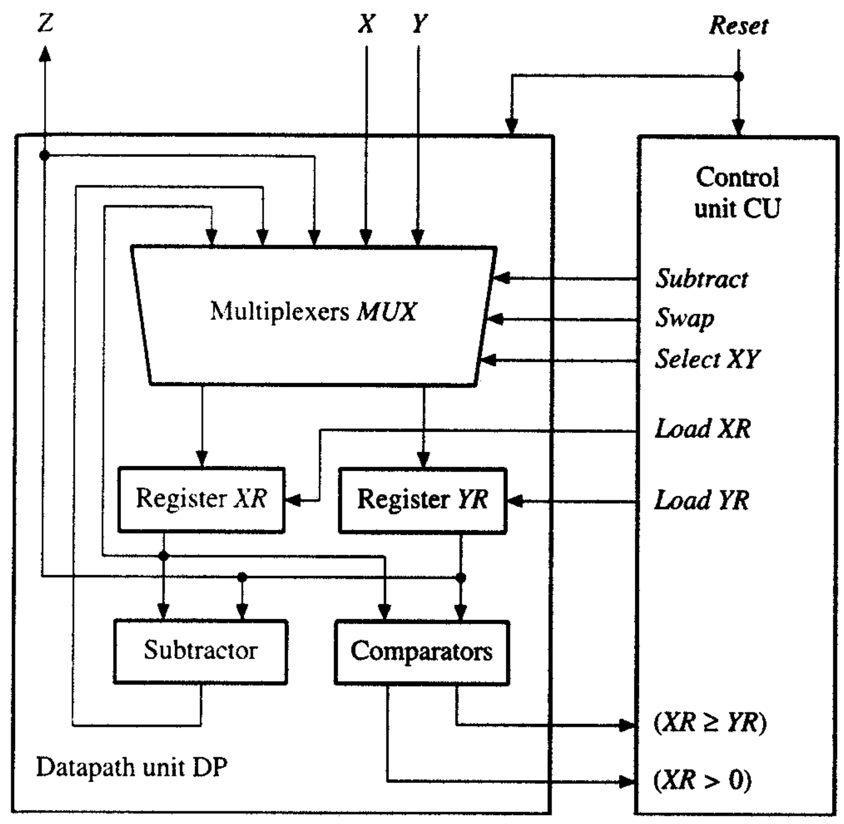
```
gcd(in: X,Y; out: Z);
    register XR, YR, TEMPR;
    XR := X;                          {Input the data}
    YR := Y;
    while XR > 0 do begin
        if XR ≤ YR then begin         {Swap XR and YR}
            TEMPR := YR;
            YR := XR;
            XR := TEMPR; end
            XR := XR – YR;            {Subtract YR from XR}
    end
    Z := YR;                          {Output the result}
end gcd;
```

Figure 5.5
Procedure *gcd* to compute the greatest common divisor of two numbers.

An Example

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

Hardware for the GCD processor



State Table for the Control Unit

State	Inputs ($XR > 0$) ($XR \geq YR$)			Outputs				
	0–	**10**	**11**	***Subtract***	***Swap***	***Select XY***	***Load XR***	***Load YR***
S_0 (Begin)	S_3	S_1	S_2	0	0	1	1	1
S_1 (Swap)	S_2	S_2	S_2	0	1	0	1	1
S_2 (Subtract)	S_3	S_1	S_2	1	0	0	1	0
S_3 (End)	S_3	S_3	S_3	0	0	0	0	0

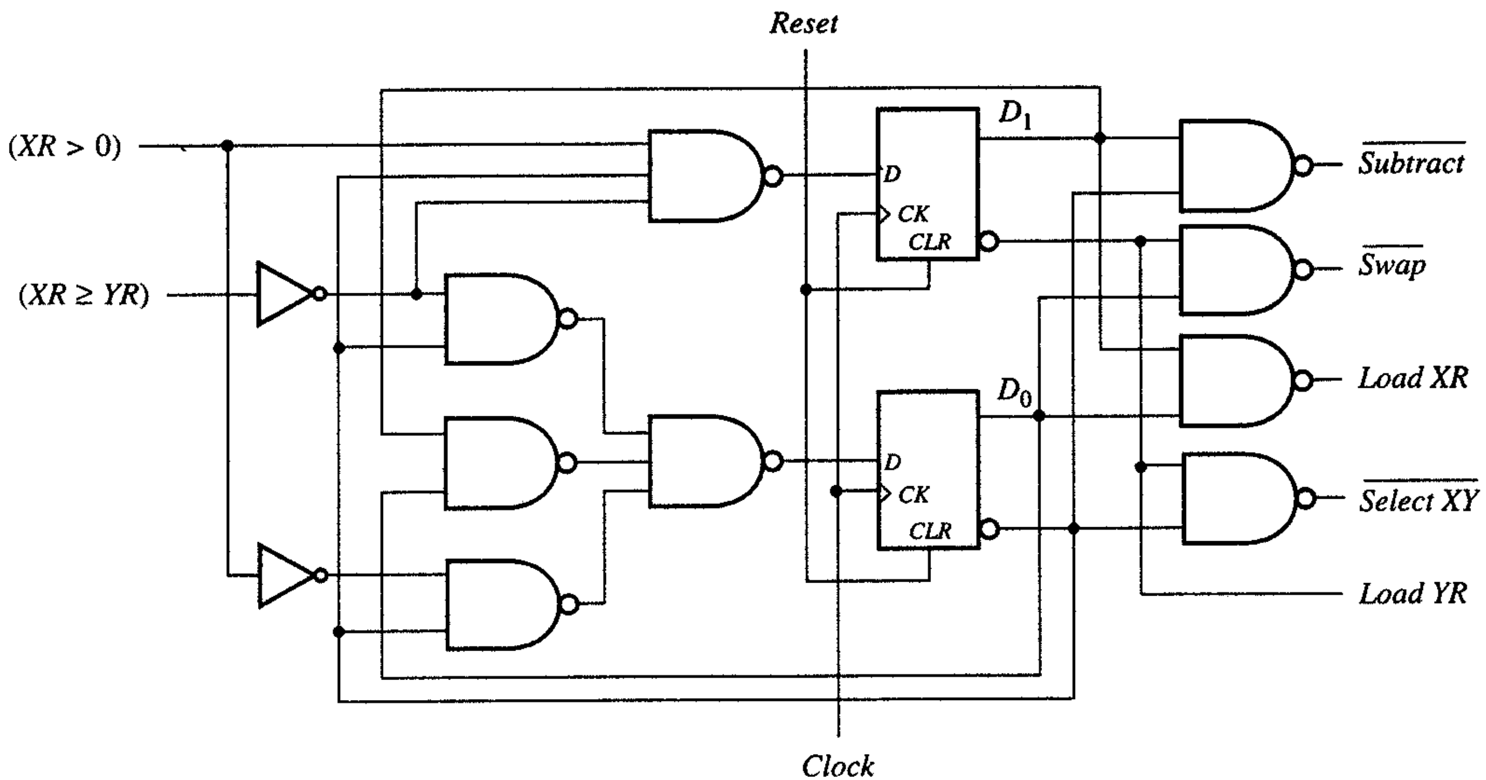
What kind of state machine is this?

Classical method

$S_0 = 00$, $S_1 = 01$, $S_2 = 10$ and $S_3 = 11$

Inputs		Present state		Next state		Outputs				
$(XR > 0)$	$(XR \geq YR)$	D_1	D_0	D_1^+	D_0^+	*Subtract*	*Swap*	*Select XY*	*Load XR*	*Load 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

Excitation Table



Is this a Moore or Mealy Machine?

Design based on Microprogram

Concept of Microprogram

- High Level description of a double precision ADD:
  - ADD AL, BL
  - ADDC AH, BH
- Low level description: Microprogram

Cycle	Function Select	Storage Control	Data Routing
1	Add	Read AL, Read BL, Write AL	…
2	Add with carry	ReadAH, Read BH, Write AH	…

What is a Microprogram?

Microprogram

- Program stored in memory that generates all the control signals required to execute the instruction set correctly

- Consists of microinstructions

Microinstruction

- Contains a control word and a sequencing word

Control Word - All the control information required for one clock cycle

Sequencing Word - Information needed to decide the next microinstruction address

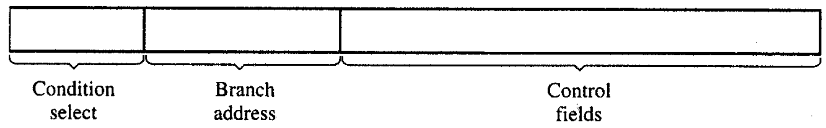
Control Memory(Control Storage: CS)

- Storage in the microprogrammed control unit to store the microprogram

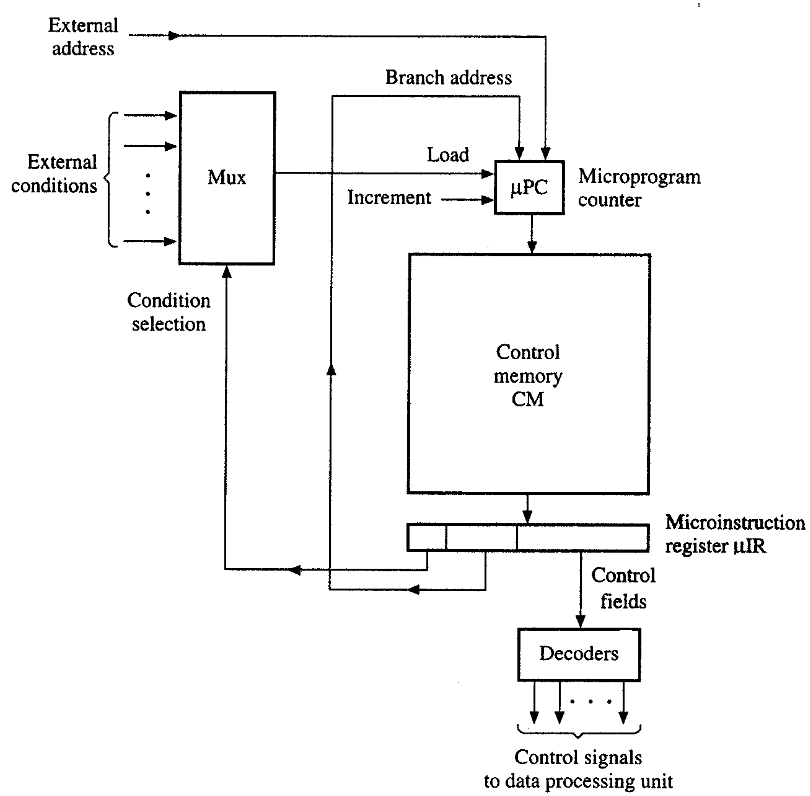
What is a Microprogram?

- *Microprogramming* is so named because it uses concepts from regular *programming*. The *micro* prefix should remind you, however, that the microprogram used by a processor is different from the program executed by the processor.
- The main thing to remember is that we have a computer inside a computer, but that the inner computer is much simpler and more restricted

Micro-programmed Control Unit



(a)



Symbolic Micro-program

Begin: A=0, Count=0, F=0, M=InBus → c_9, c_{10}

Input: Q=Inbus → c_8

Test1: If Q[0]=0 then goto RSHIFT;

Add: A[7:0]=A[7:0]+M[7:0], F=(m[7] and Q[0])or F → c_2,c_3,c_4

Rshift: A[7]=F, A[6:0],Q=A,Q[7:1], Cnt=Cnt+1

if cnt≠7 then goto Test1 → c_0,c_1,c_{11}

Test2: If Q[0]=0 then go to Output1

Subtract: A[7:0]=A[7:0]-M[7:0], Q[0]=0 → c_2,c_3,c_4,c_5

Output1: Outbus=A → c_6

Output2: Outbus=B → c_7

End: Halt → END

Control Signals

Control Signal	Operation controlled
c_0	Set sign bit of A to F.
c_1	Right-shift register-pair A.Q.
c_2	Transfer adder output to A.
c_3	Transfer A to left input of adder.
c_4	Transfer M to right input of adder.
c_5	Perform subtraction (correction). Clear Q[0].
c_6	Transfer A to output bus.
c_7	Transfer Q to output bus.
c_8	Transfer word on input bus to Q.
c_9	Transfer word on input bus to M.
c_{10}	Clear A, COUNT, and F registers.
c_{11}	Increment COUNT.
END	Completion signal (CU idle).

Branching

No Branching

Branch if Q[0]=0

Branch if Count≠7

Unconditional Branch

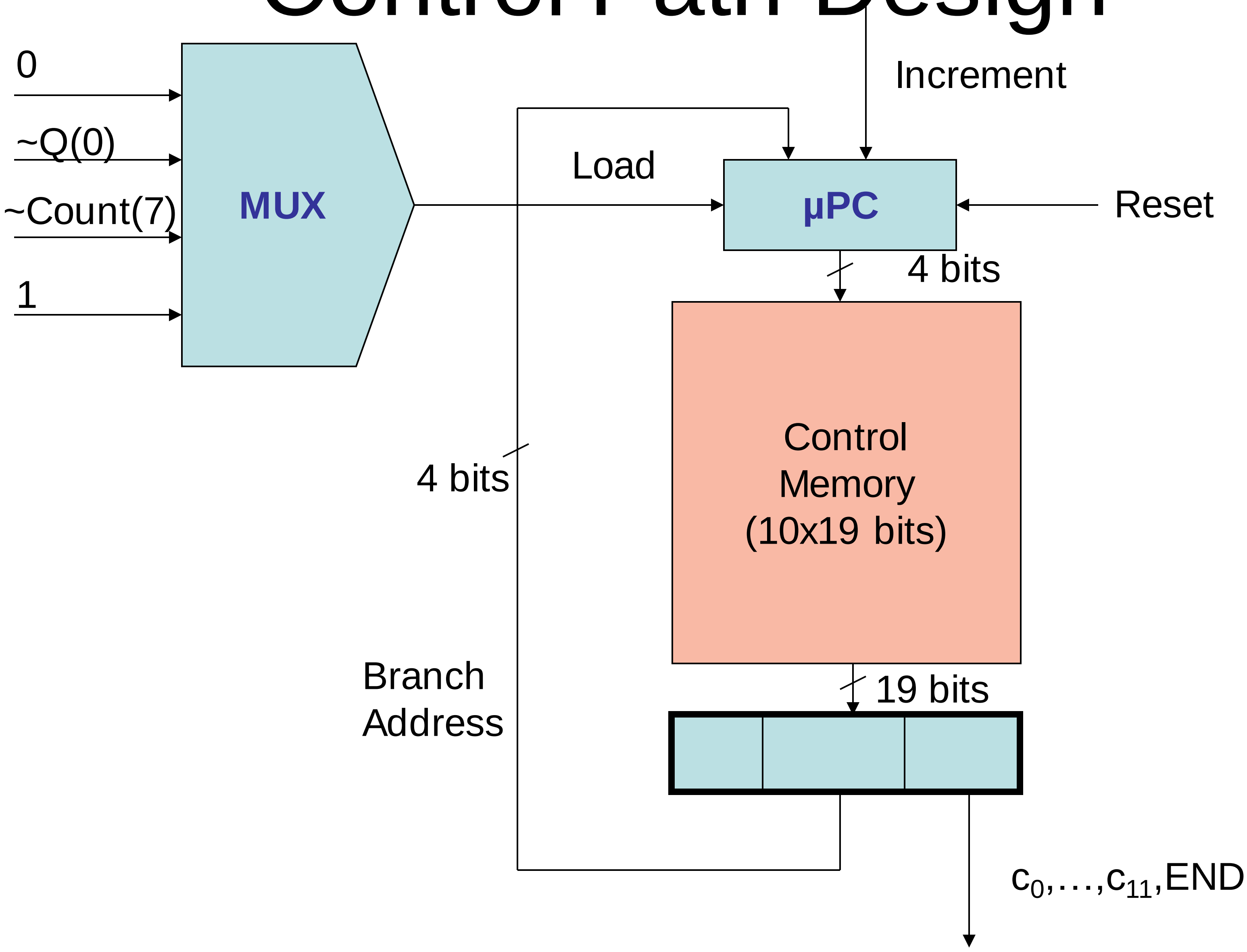
Hence a 2-bit conditional select field is needed.

There are 10 states, so 4 bits are enough to encode the states.

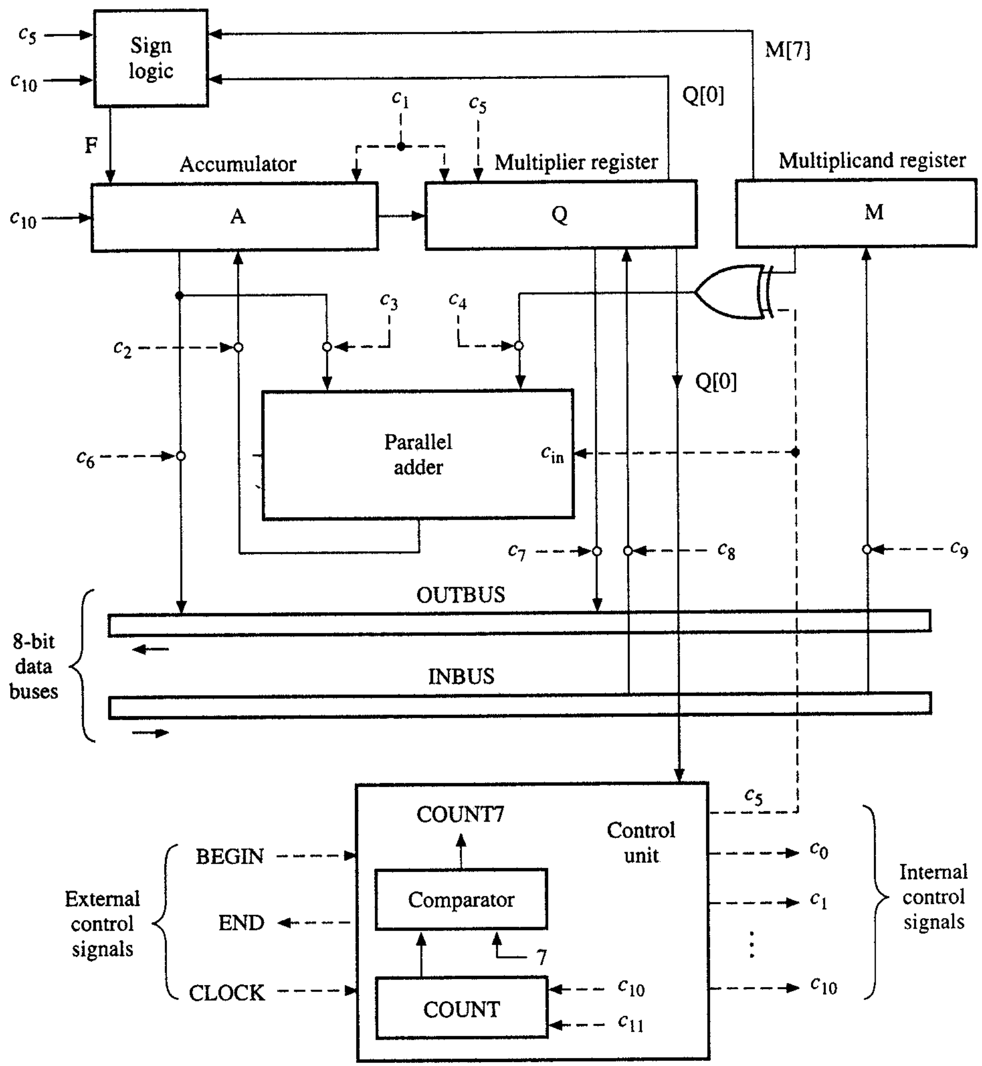
Binary Microprogram

Address in CM	Condition Selct	Branch Address	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	END
0000	00	0000	0	0	0	0	0	0	0	0	0	1	1	0	0
0001	00	0000	0	0	0	0	0	0	0	0	1	0	0	0	0
0010	01	0100	0	0	0	0	0	0	0	0	0	0	0	0	0
0011	00	0000	0	0	1	1	1	0	0	0	0	0	0	0	0
0100	10	0010	1	1	0	0	0	0	0	0	0	0	0	1	0
0101	01	0111	0	0	0	0	0	0	0	0	0	0	0	0	0
0110	00	0000	0	0	1	1	1	1	0	0	0	0	0	0	0
0111	00	0000	0	0	0	0	0	0	1	0	0	0	0	0	0
1000	00	0000	0	0	0	0	0	0	0	1	0	0	0	0	0
1001	11	1001	0	0	0	0	0	0	0	0	0	0	0	0	1

Control Path Design



Data Path Design



Comments

- Micro-programming helps in making Control Units which may be changed by changing the content of the memory.
- But slow due to the fetch timing of the instruction from the memory.

Assignment 2

1. Write a verilog code to implement the control path for a gcd processor.
2. Write a verilog code to implement the micro-programmed control unit of a 2's complement signed fraction multiplier.

Deadline: 21/3/08