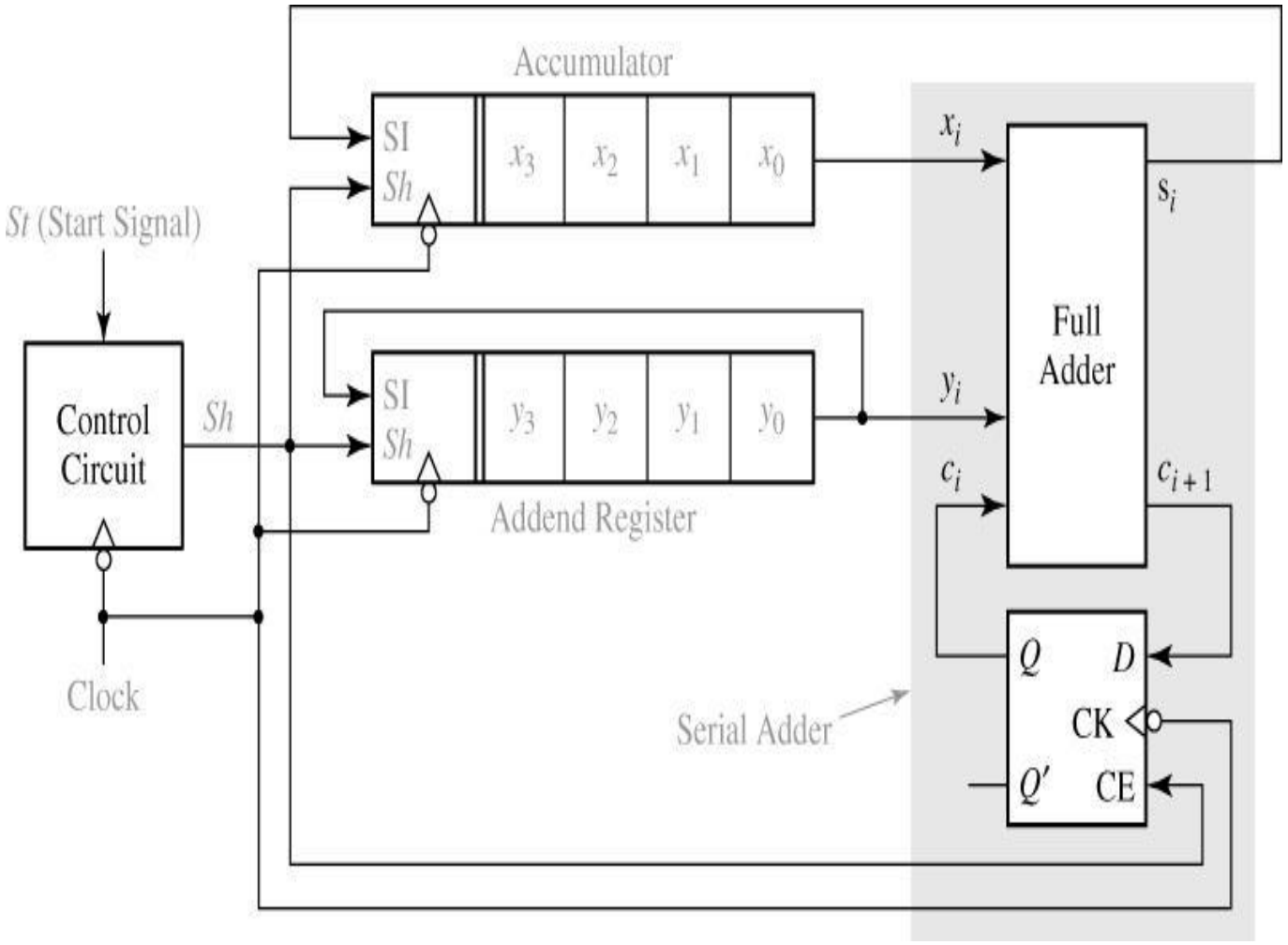
Lecture 16
Arithmetic Circuits

- Serial adder
- Parallel multiplier
- Divider
- Control circuit outputs a sequence of control signals that cause arithmetic operation to take place at appropriate times.

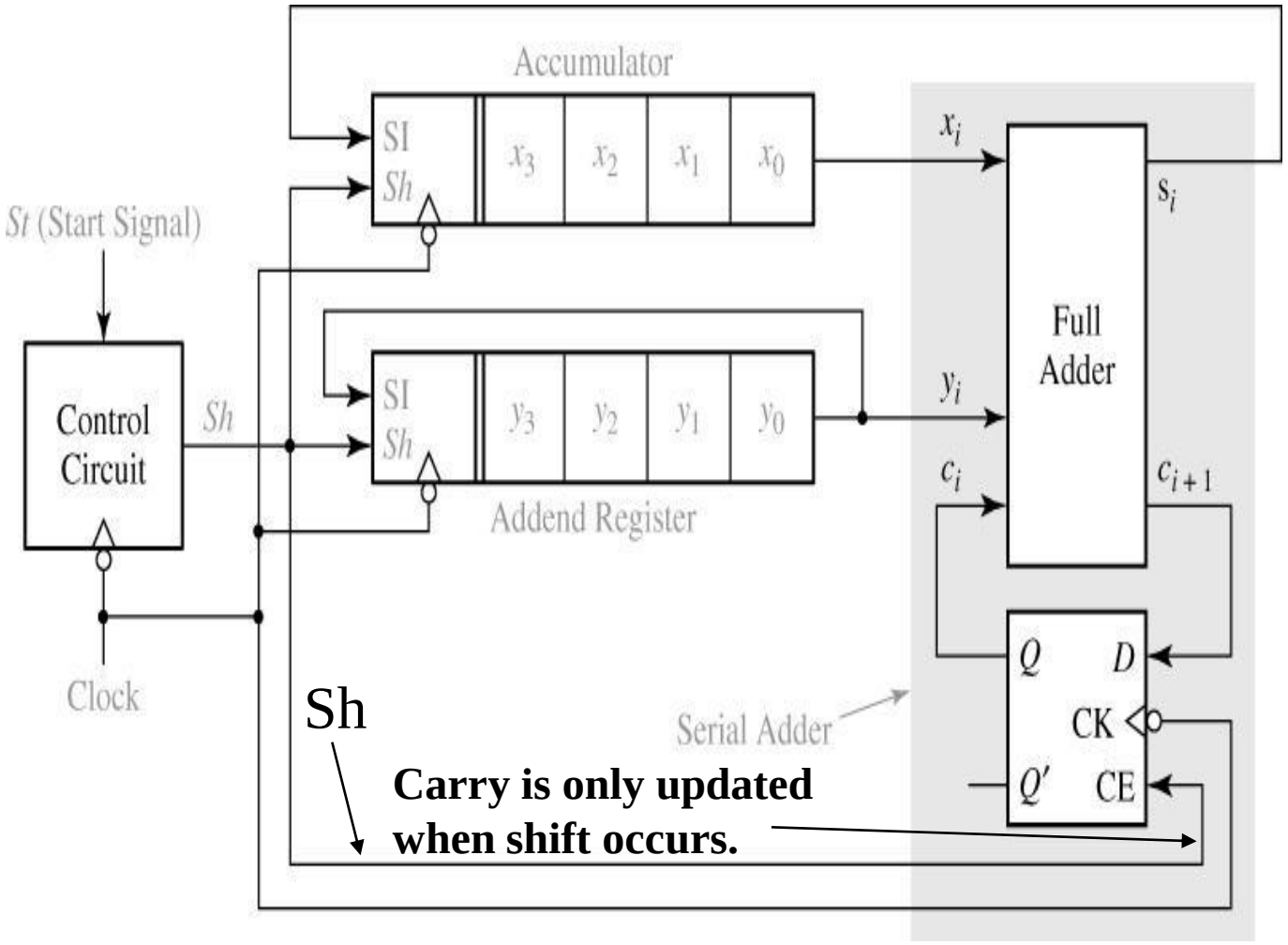
Serial Adder

- **X register**
  - Accumulator: X = X+Y
- **Y register**
  - Addend register: cyclic shift register
    - after shifting 4 times, it is back in its original state
- Sh: shift signal
- SI: serial input



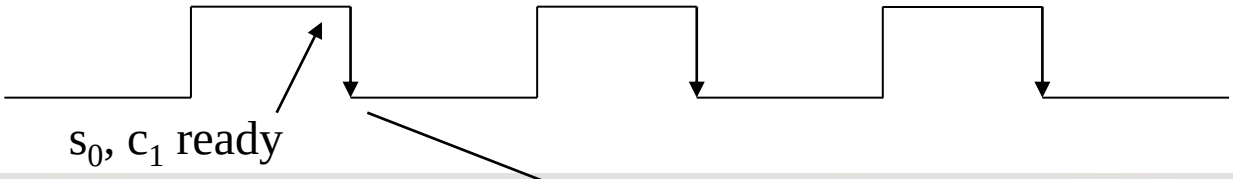
Serial Adder

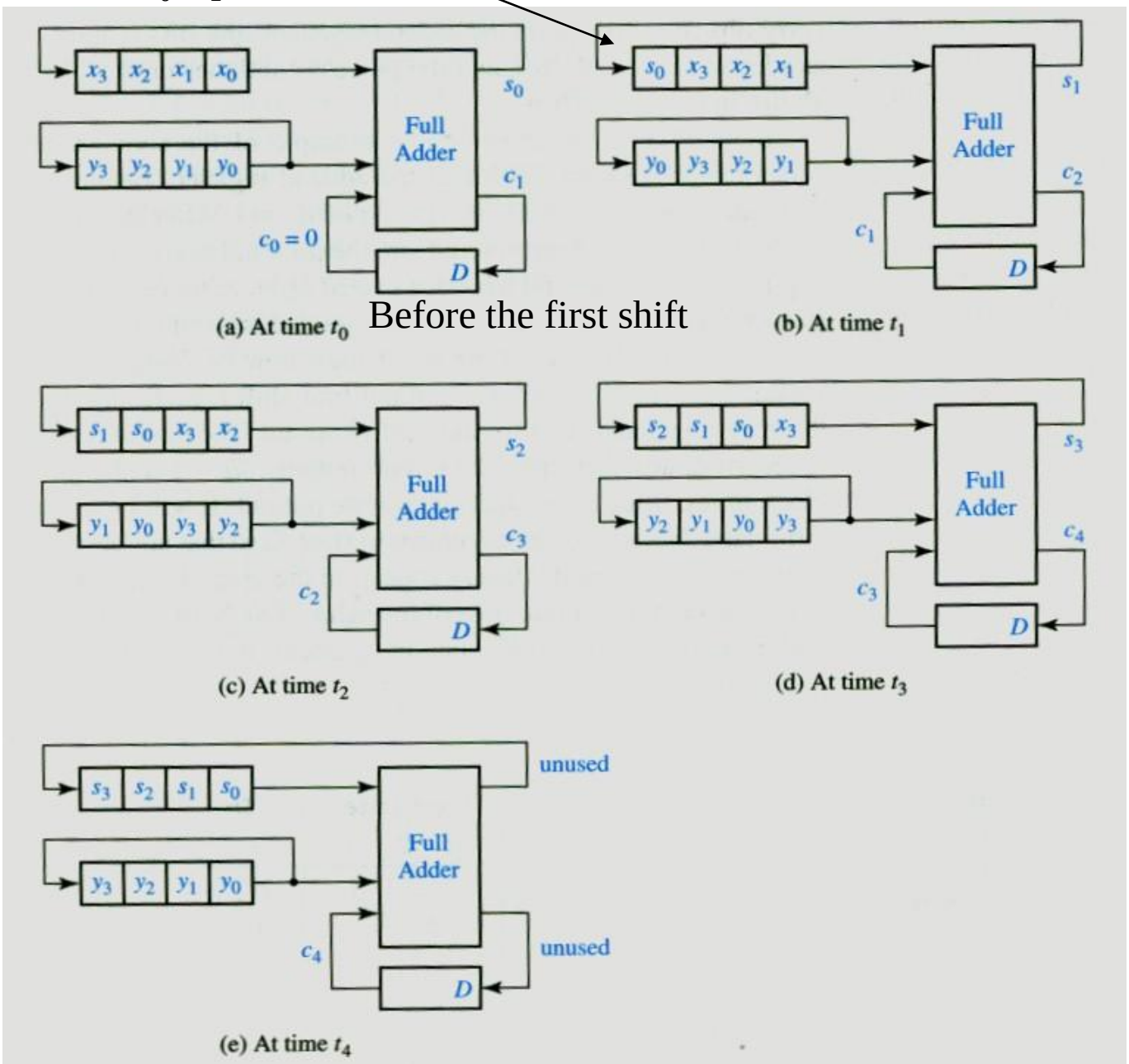
- **At the falling clock edge and Sh= 1**
  - Si (sum bi) is shifted into *x*
  - Carry bit is stored in the D FF.
  - Register *y* is rotated one bit to the right.
  - Initial loading circuit not shown.



Serial Adder

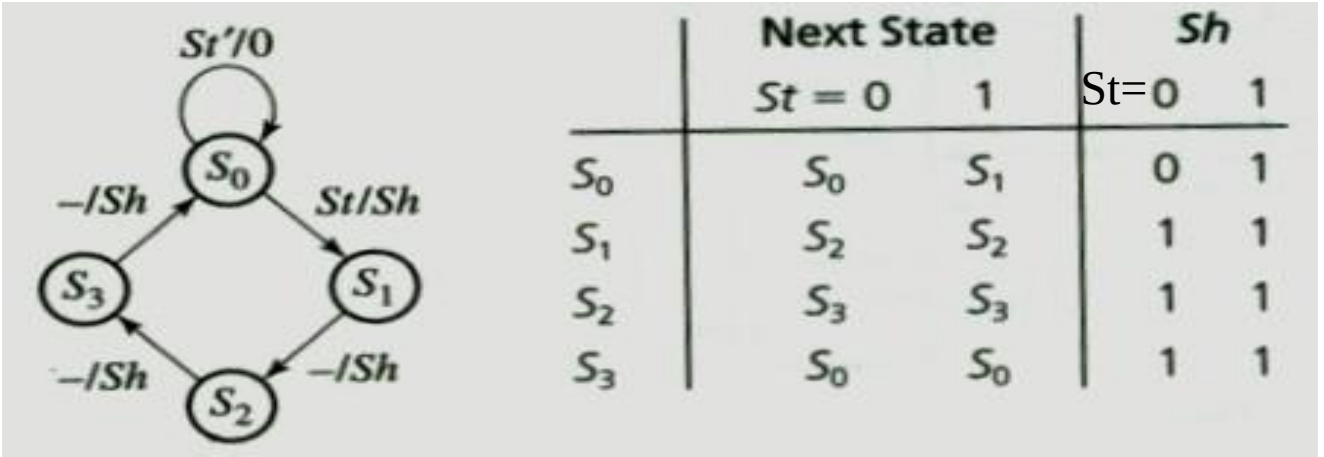
- **At the falling clock edge and Sh= 1**





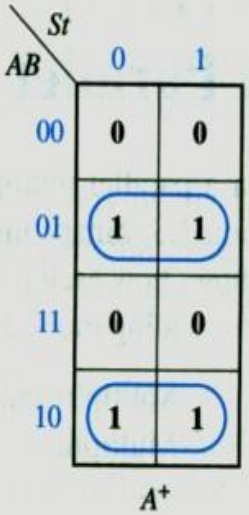
Control Unit Realization

- A start signal: St.
  - If St is asserted, then the control unit puts out 4 shift signals then stops.

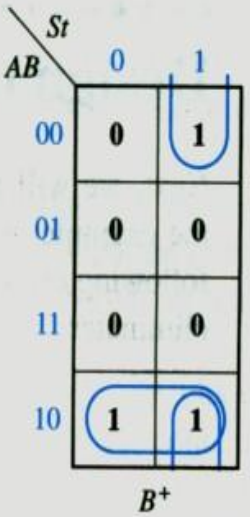


	Next State		Sh	
	St = 0	1	St=0	1
S_0	S_0	S_1	0	1
S_1	S_2	S_2	1	1
S_2	S_3	S_3	1	1
S_3	S_0	S_0	1	1

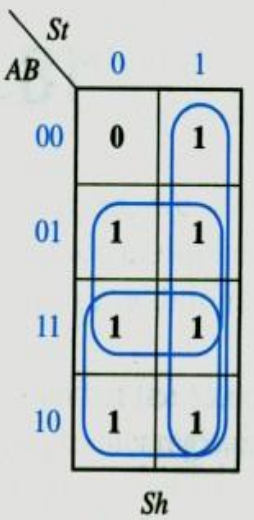
	AB	A^+B^+	
		0	1
S_0	00	00	01
S_1	01	10	10
S_2	10	11	11
S_3	11	00	00



$D_A = A'B + AB'$
$= A \oplus B$



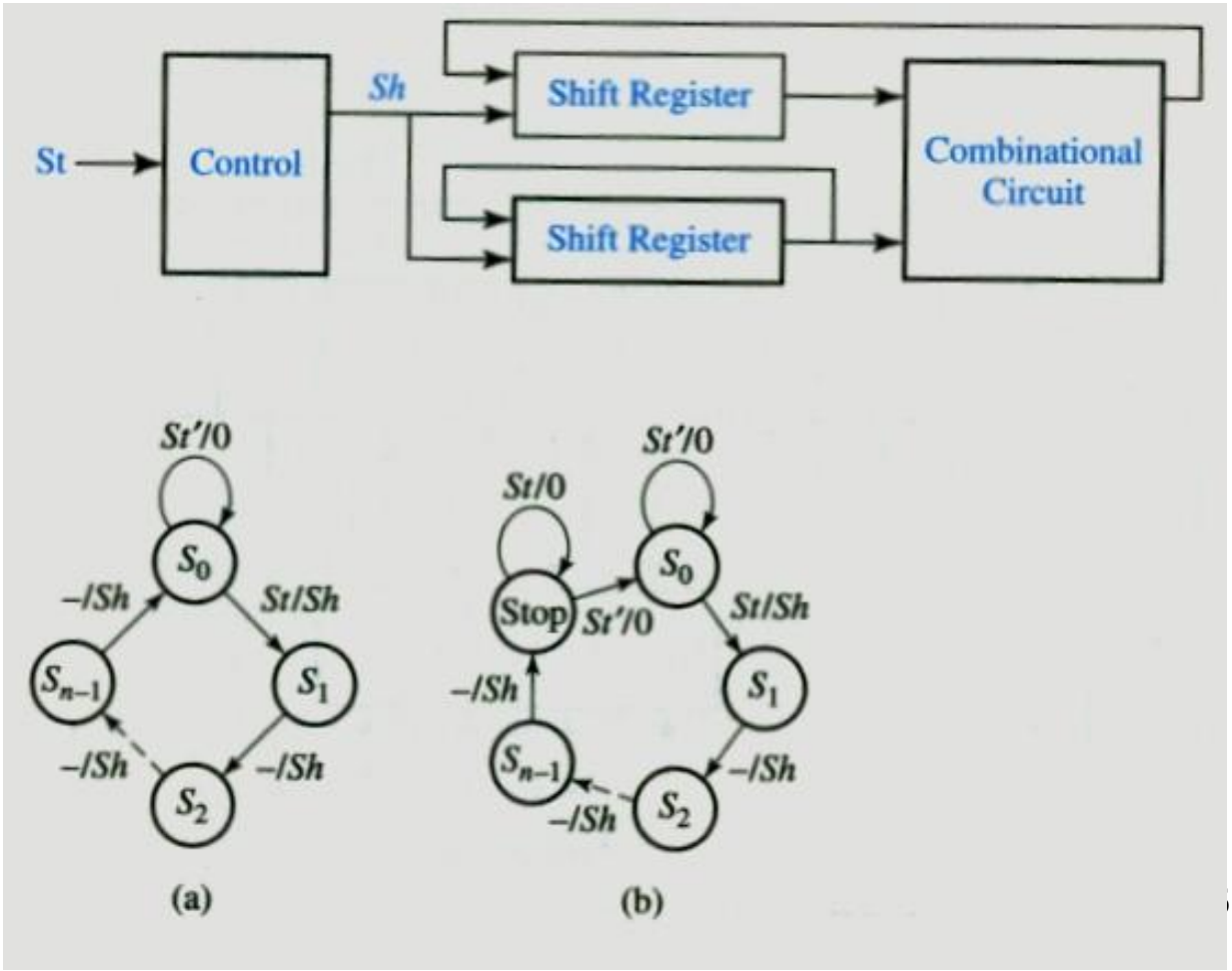
$D_B = St\,B' + AB'$



$Sh = St + A + B$

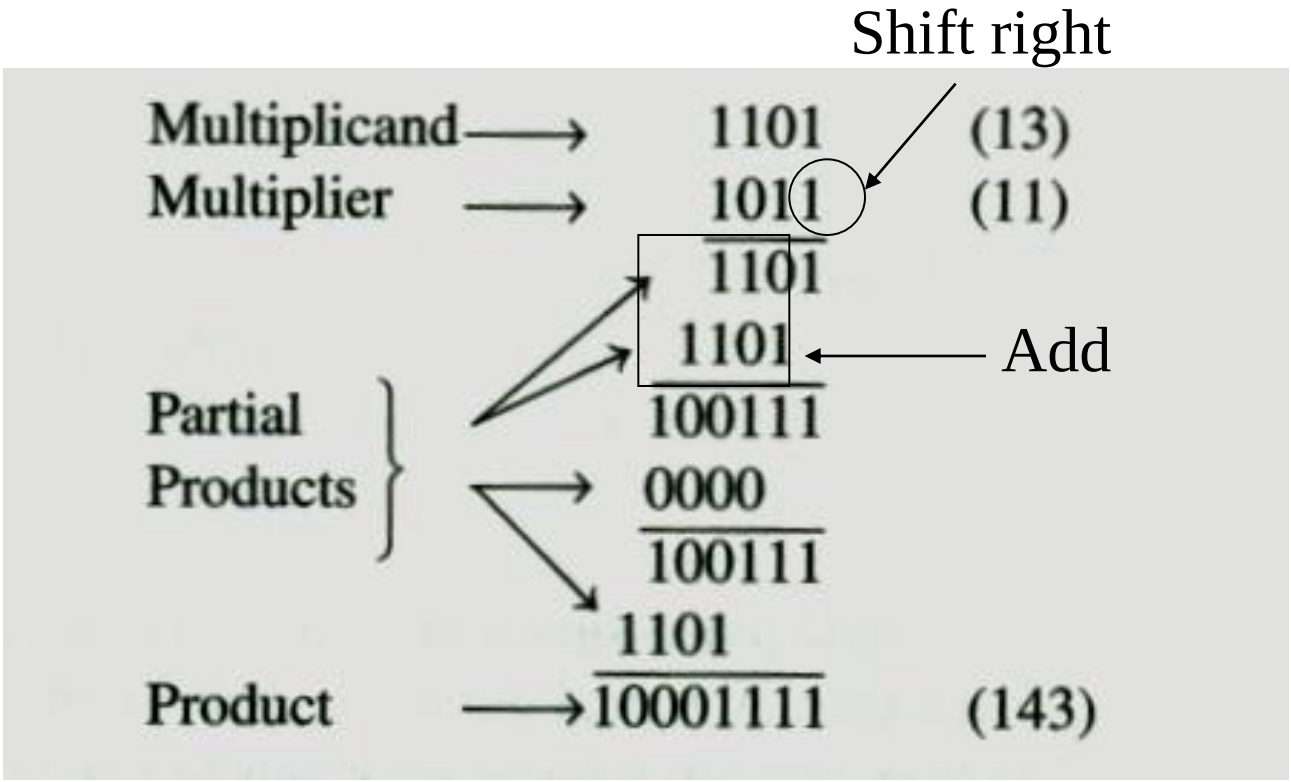
Serial Processing Unit

- A start signal: St.
  - If St is asserted, then the control unit puts out *n* shift (Fig. a) St = 1 for only one clock time.
  - If St remains 1 until the shifting is completed, then a stop state is required. (Fig. b)



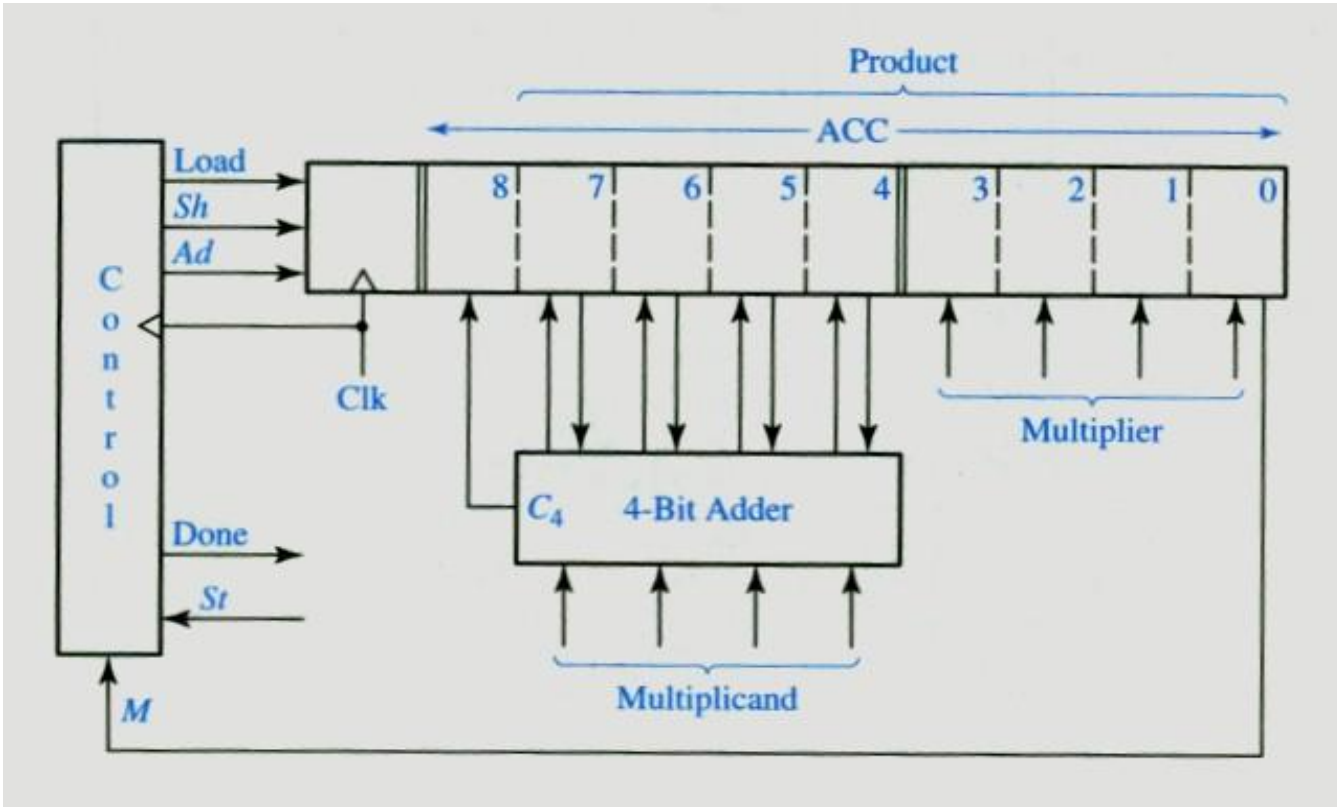
Multiplier

- Add shift add shift add….
- Check the right most bit of the multiplier. 1=> add + shift, 0=> shift.



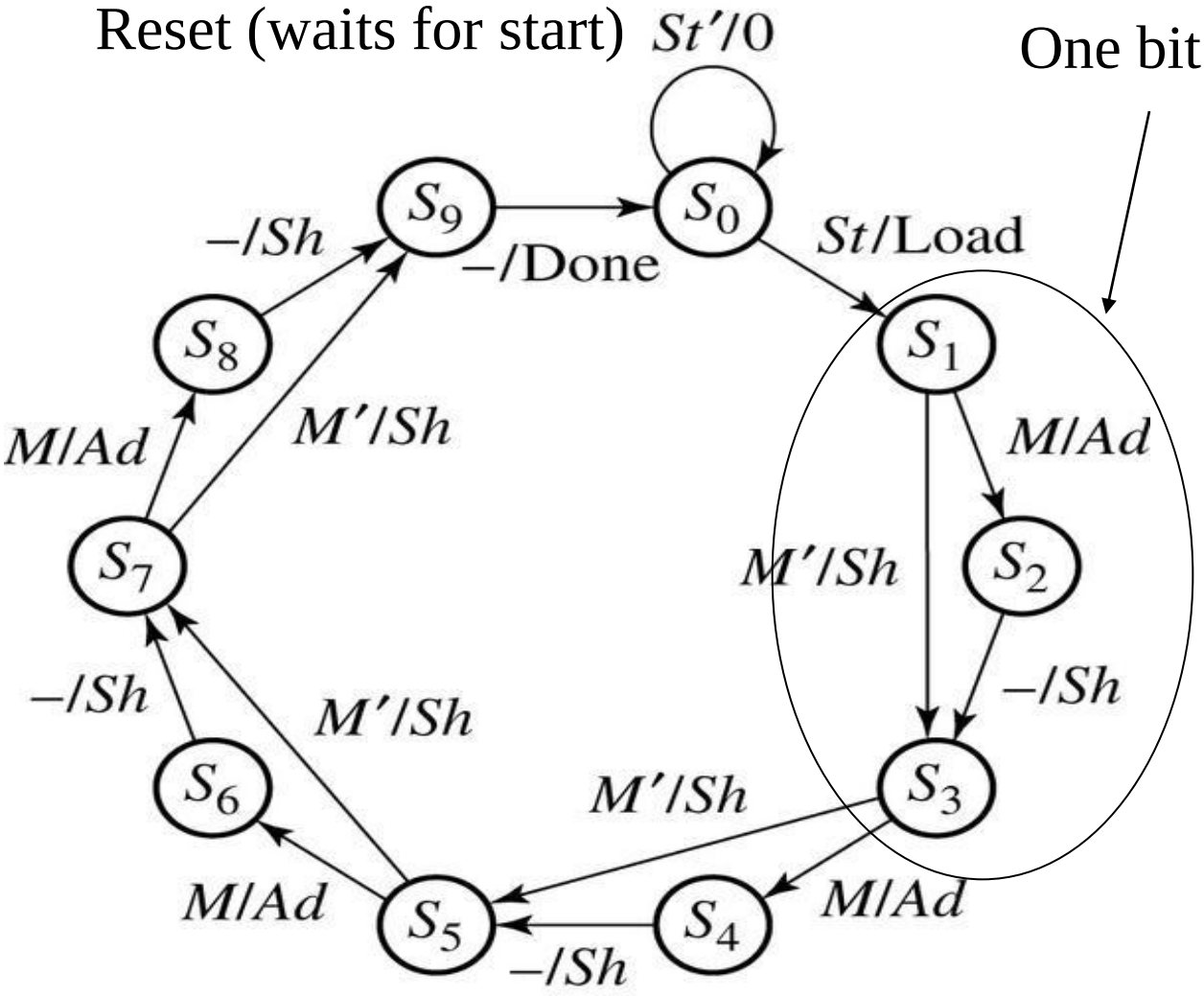
Multiplier Diagram

- Product register [3:0] = multiplier
- Load: ACC = 00000::multiplier
- Sh: shift ACC 1 bit right.
- St: start signal, start operation.
- M bit: M= 1, add + shift. M = 0, shift.



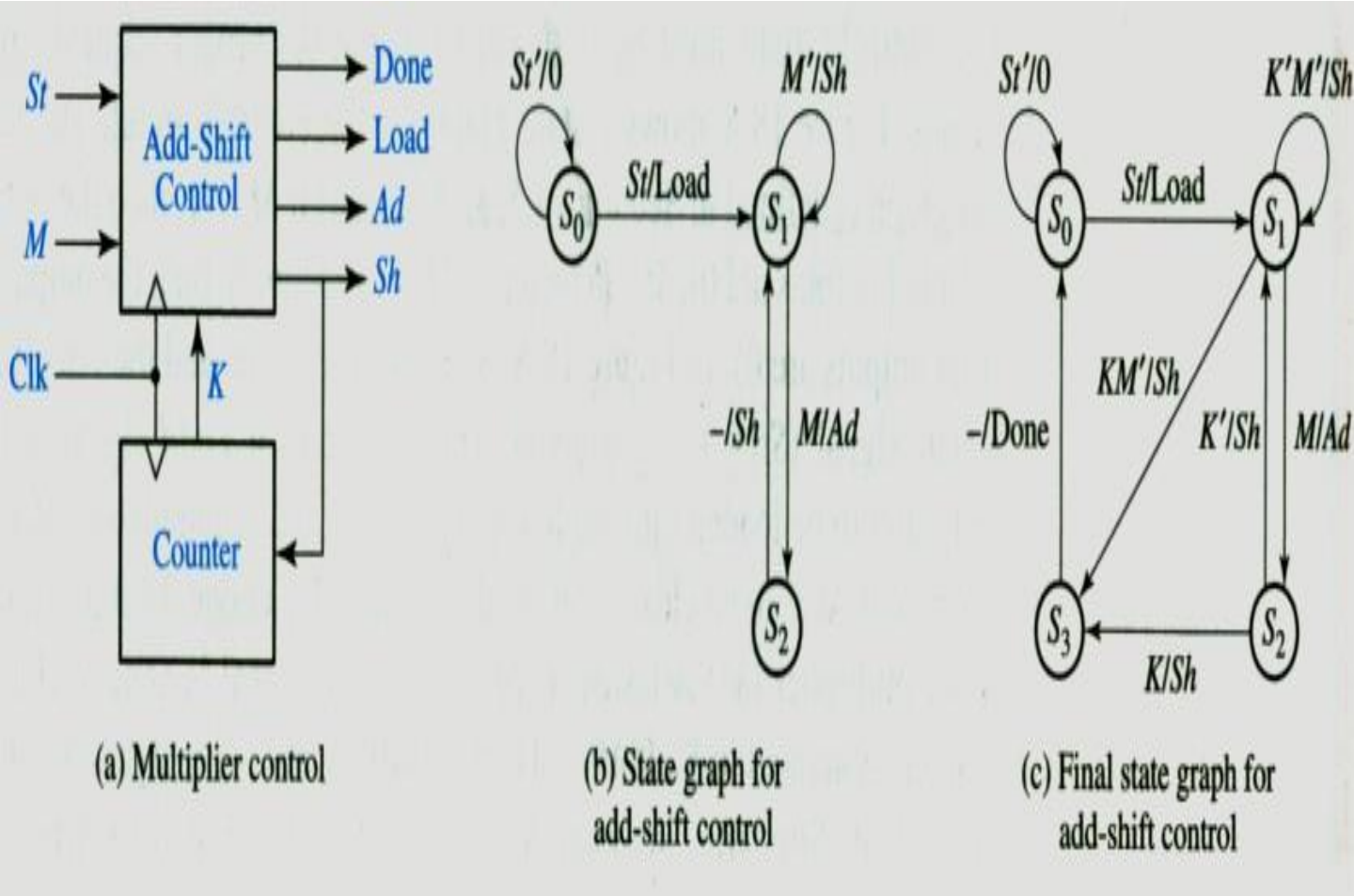
Multiplier Control

- M/Ad: If M = 1 then Ad = 1, the rest of outputs is 0.
- How many clocks are required for a multiplication?



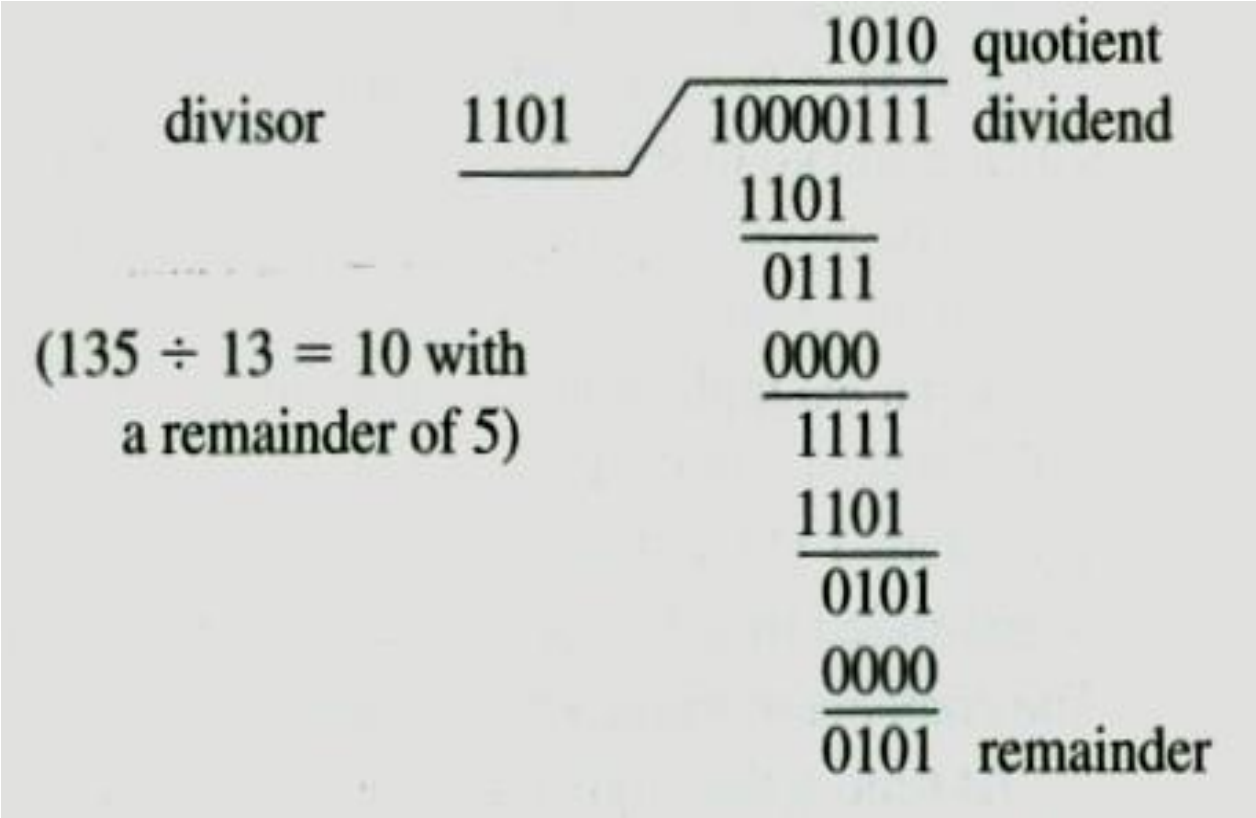
Multiplier Control

- Multiplication for a large number of bits
- User a counter. When n-1 shifts have occurred, K = 1.Then go to S3 for completion (last shift also).



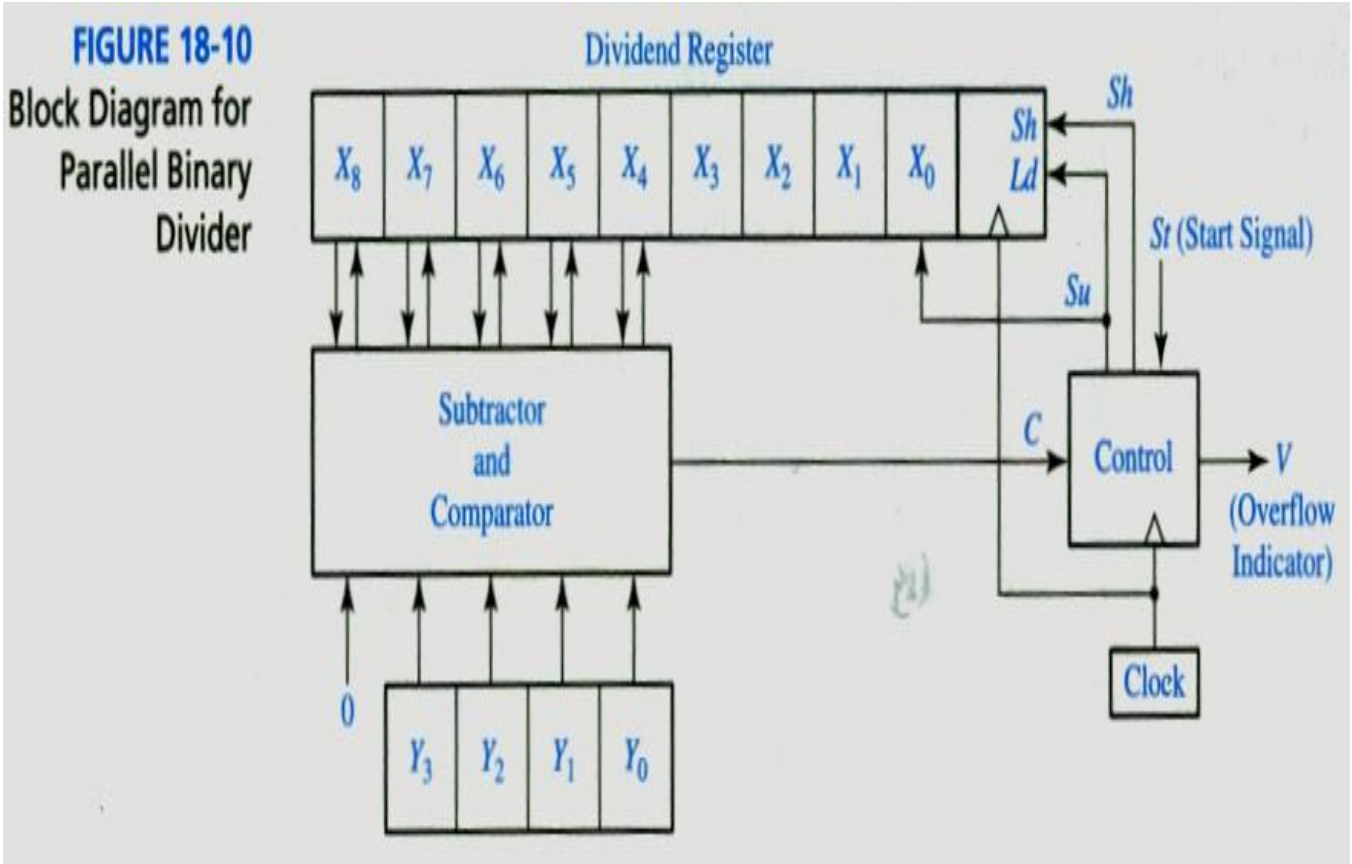
Division

- Division = subtraction and shift
- 8-bit dividend by a 4-bit divisor to get a 4-bit quotient
- Use a 9-bit dividend register for shifting the dividend left.
- Store quotient bit by bit into the dividend register when shifting left.



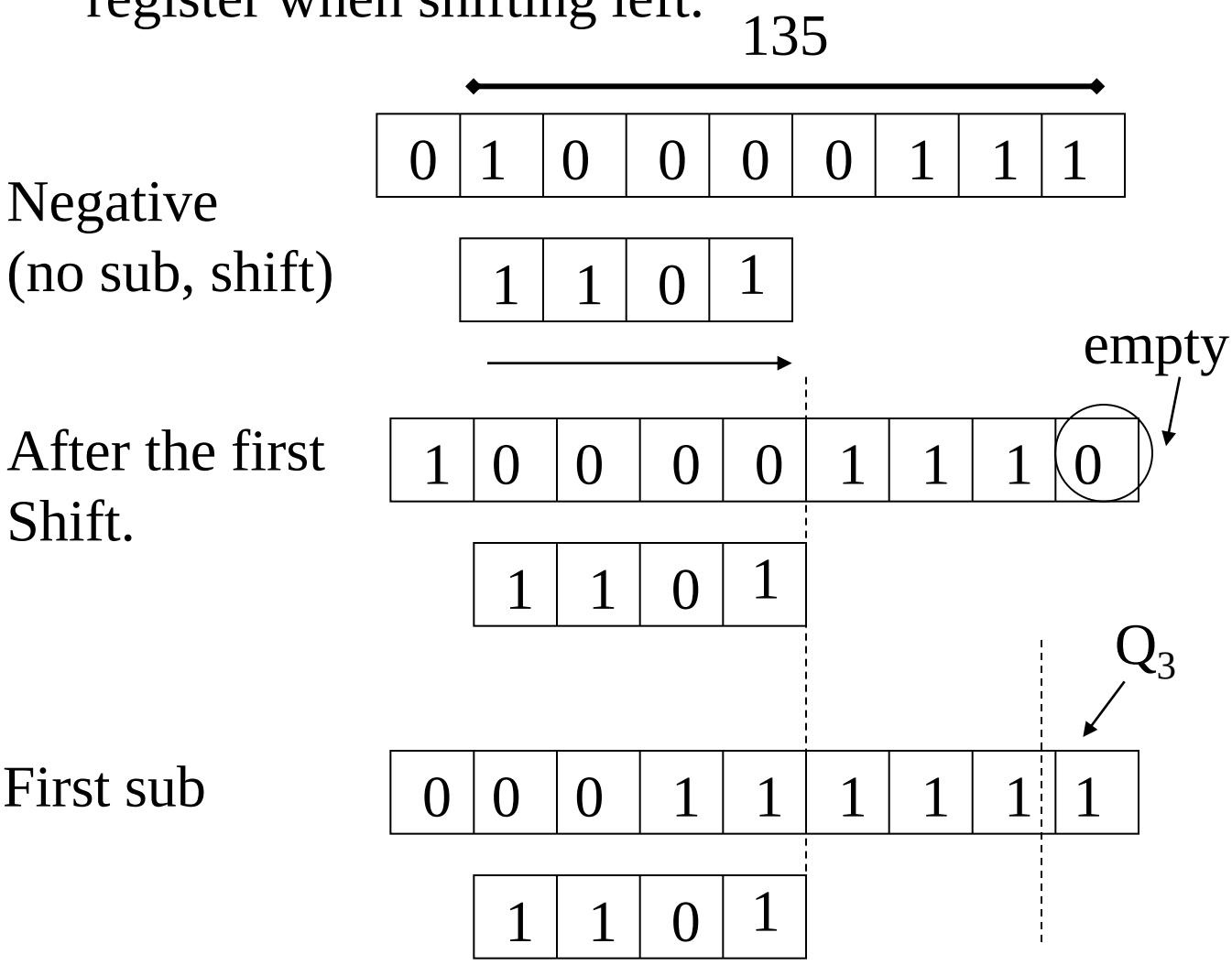
Binary Division

- Use a 9-bit dividend register for shifting the dividend left.
- Store quotient bit by bit into the dividend register when shifting left.



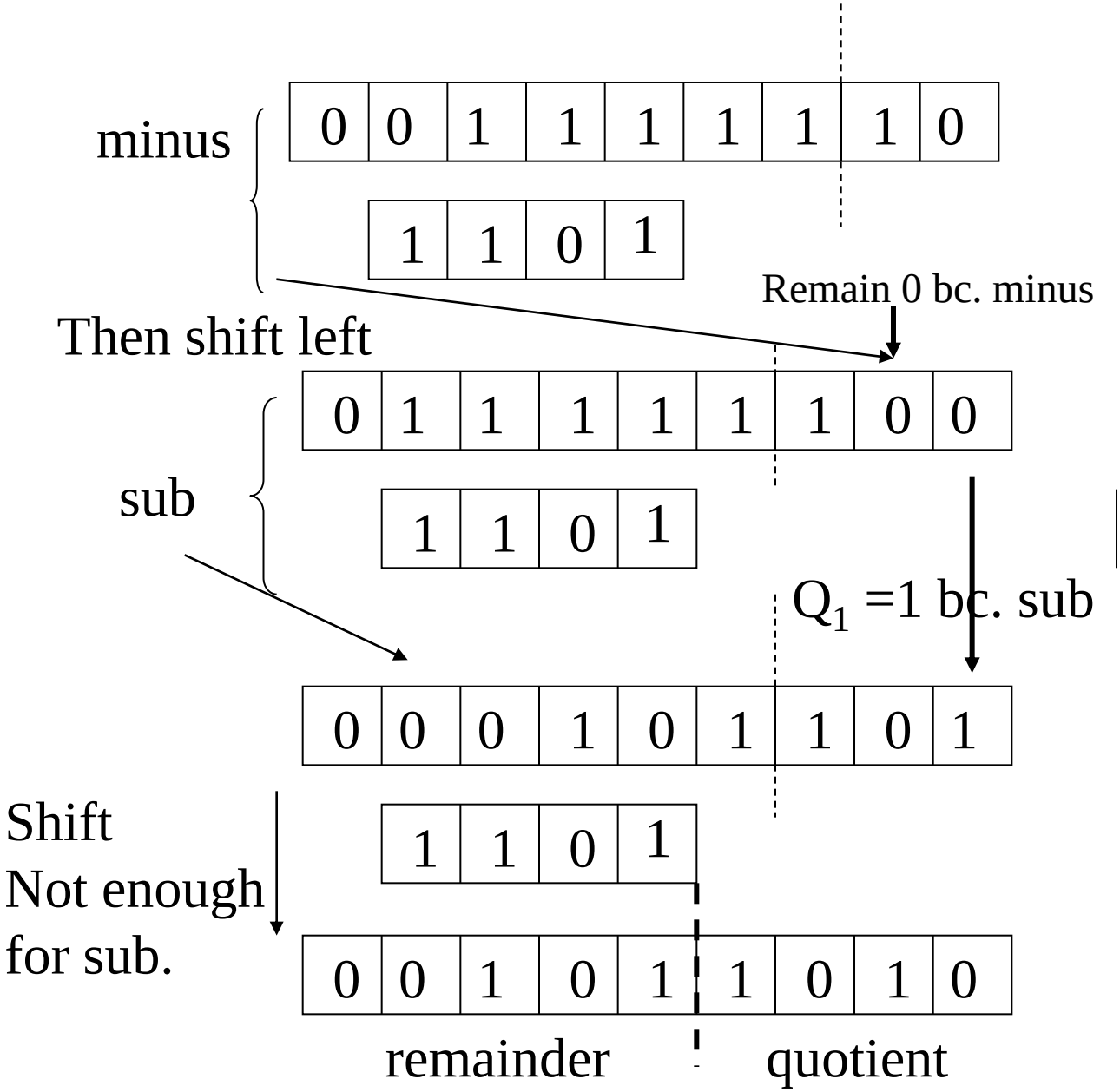
Binary Division

- Use a 9-bit dividend register for shifting the dividend left.
- Store quotient bit by bit into the dividend register when shifting left.



Binary Division (cont.)

- Use a 9-bit dividend register for shifting the dividend left.



Binary Division (cont.)

- In this example, if Q > 15, then overflow occurs.
- If initially $X_8X_7X_6X_5X_4 >= Y_3Y_2Y_1Y_0$, then the quotient will be greater than 15 and overflow occurs.
- Shift signal: Sh
  - Shift the dividend one place to the left on the next rising clock edge.
- Subtract signal:Su
  - Use a subtracter to compute $X_8X_7X_6X_5X_4$ - $Y_3Y_2Y_1Y_0$ (this is combinational circuit).
  - Su loads the subtracter output into $X_8X_7X_6X_5X_4$ and sets the quotient bit to 1 on the next rising edge.

Binary Division (cont.)

- Sh: shift the dividend register 1 bit left.
- Su: load X_8 to X_4 for the result of subtraction, (Ld). Set X_0 to 1.

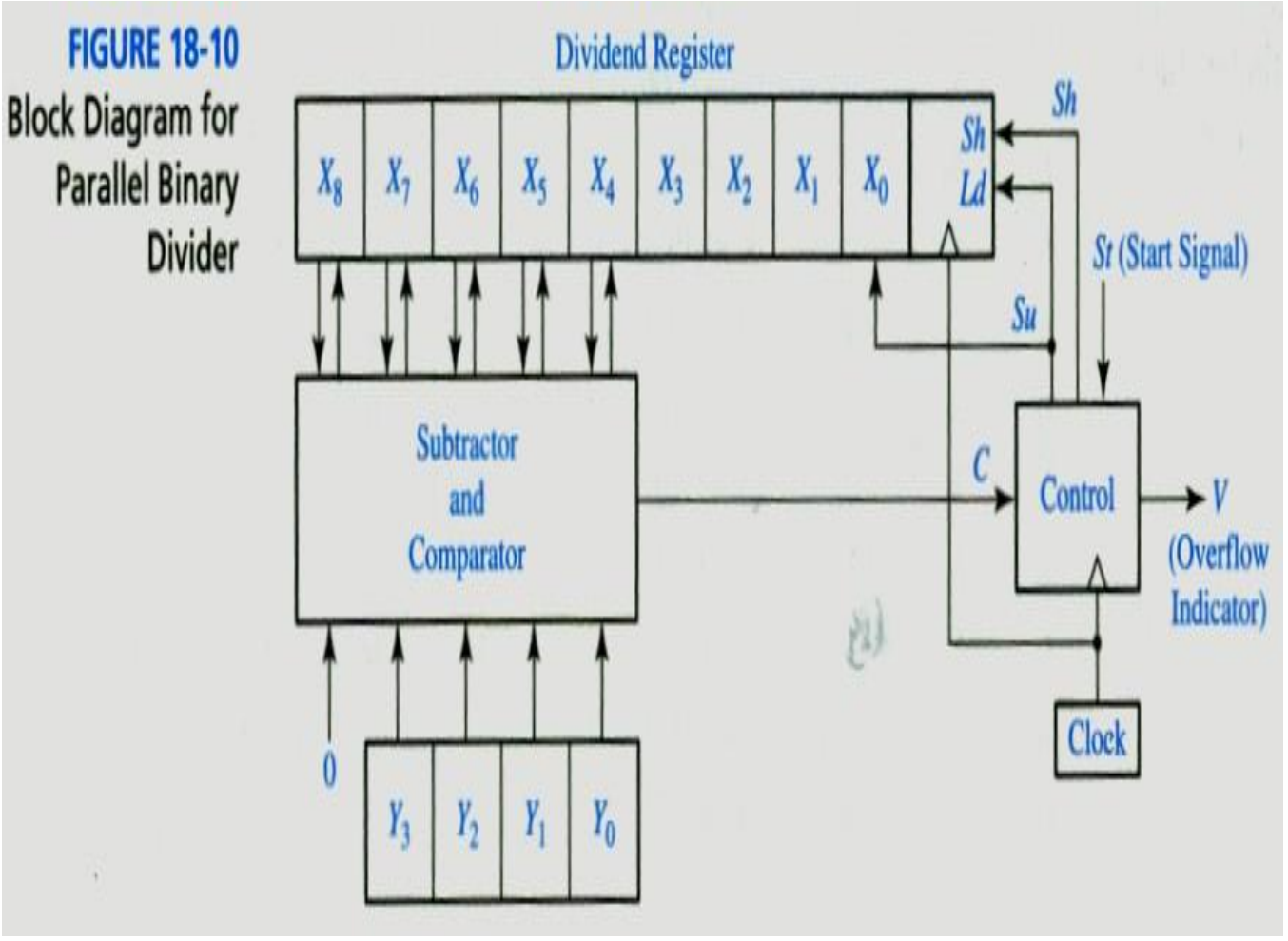
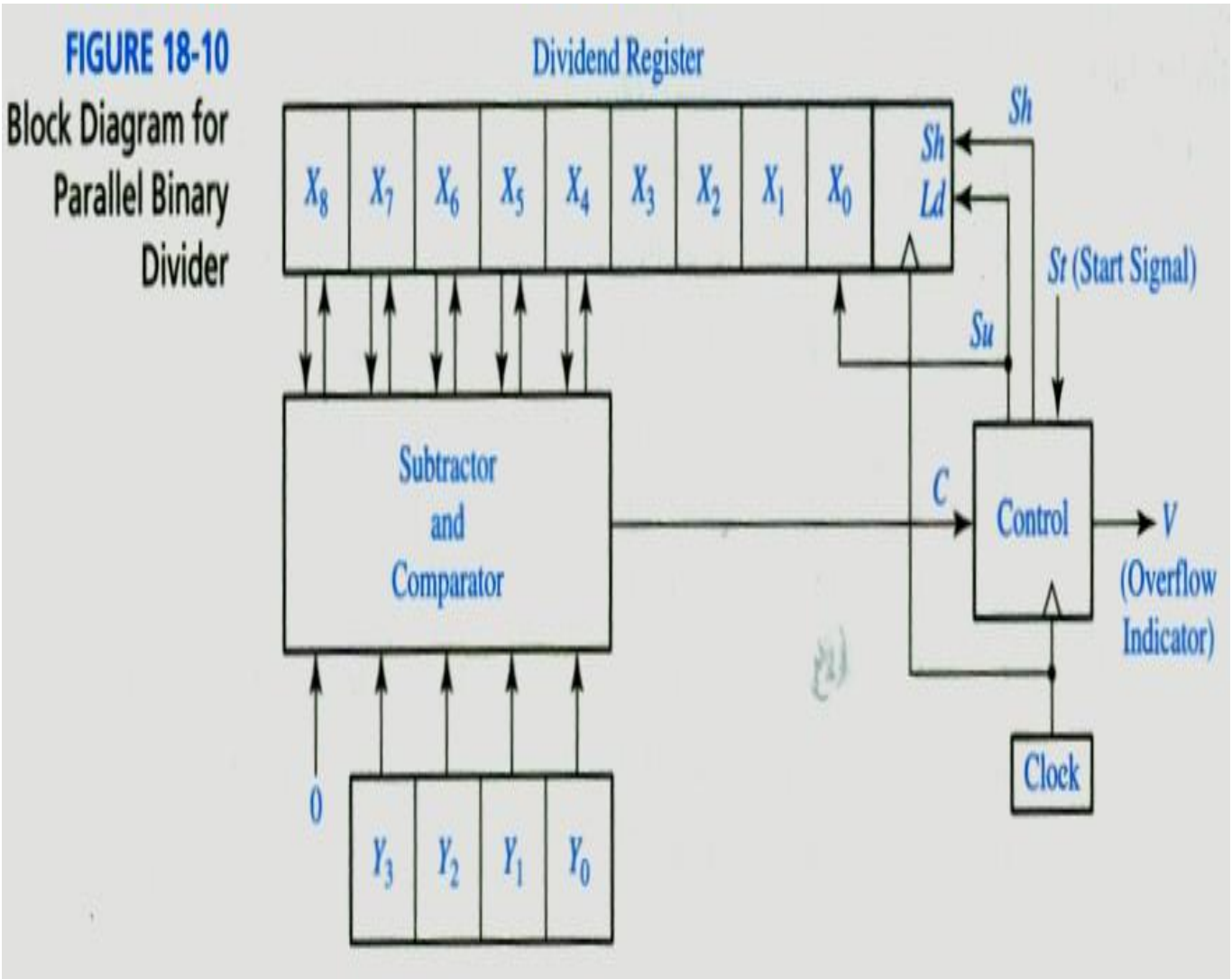


FIGURE 18-10
Block Diagram for Parallel Binary Divider

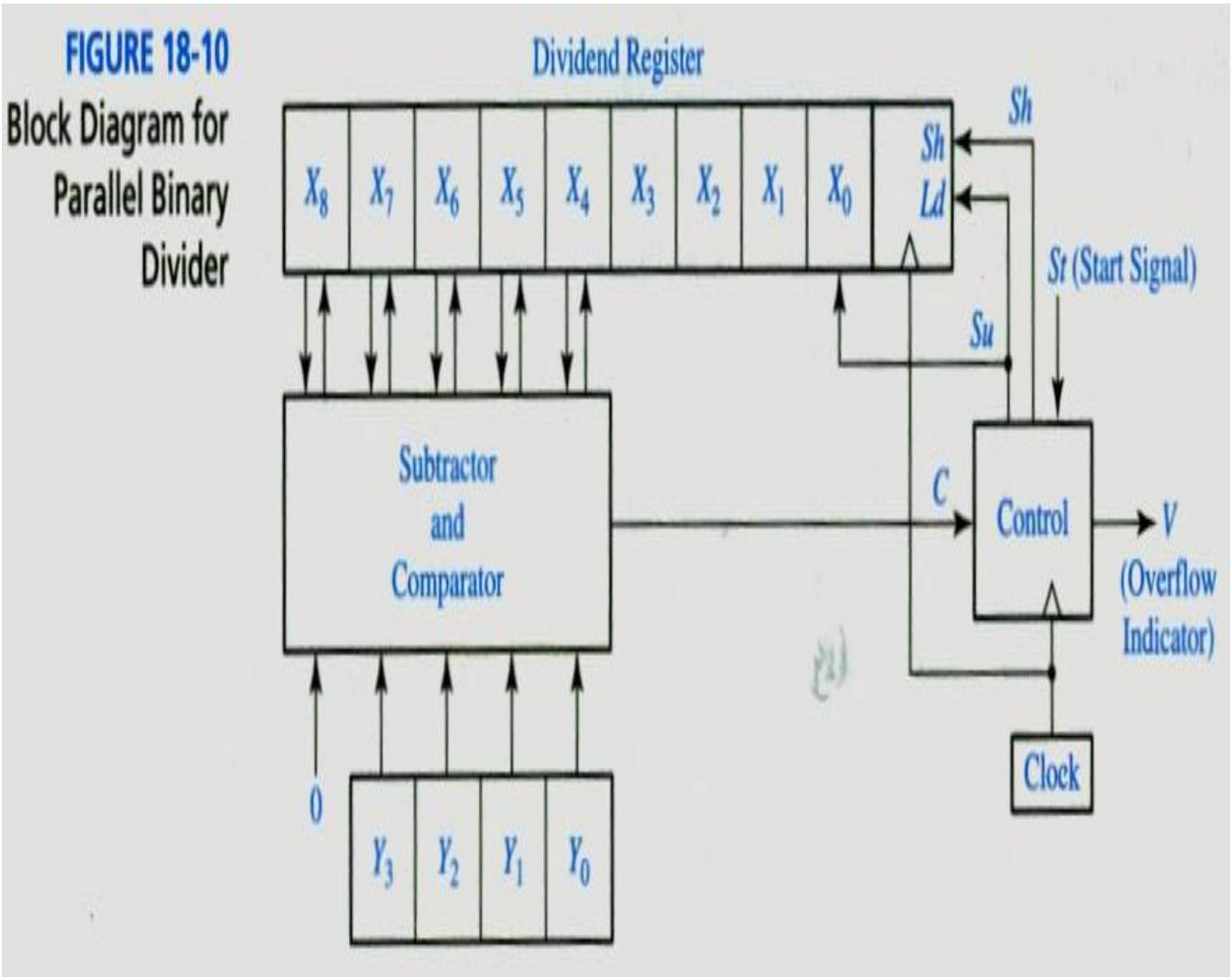
Binary Division (cont.)

- C: if divisor > X_8 to X_4, C=0. No subtraction, generate shift. Sh=1.
- Otherwise C =1, subtract signal is generated (Su =1), Q bit is set to 1.



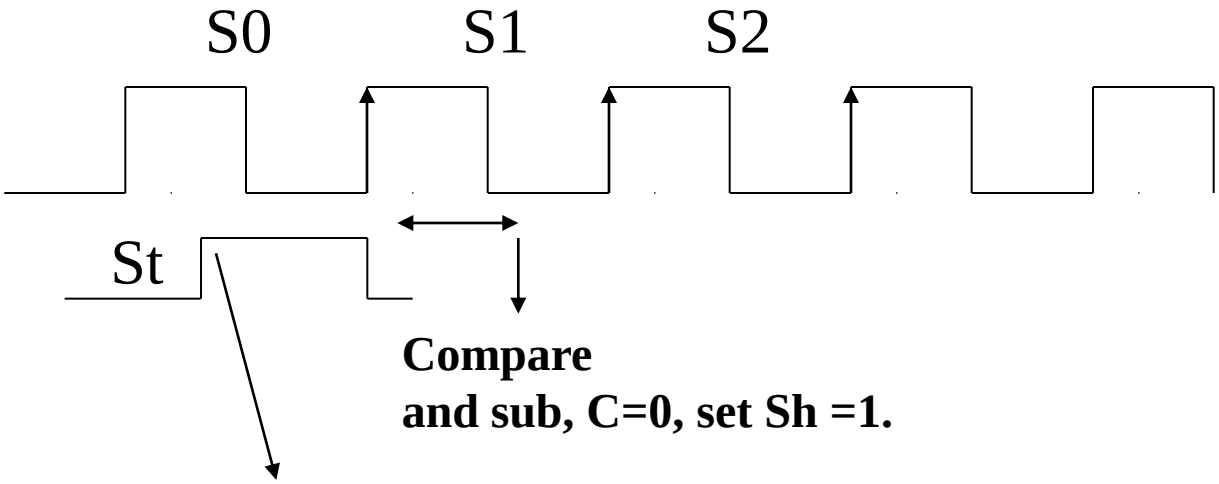
Binary Division (cont.)

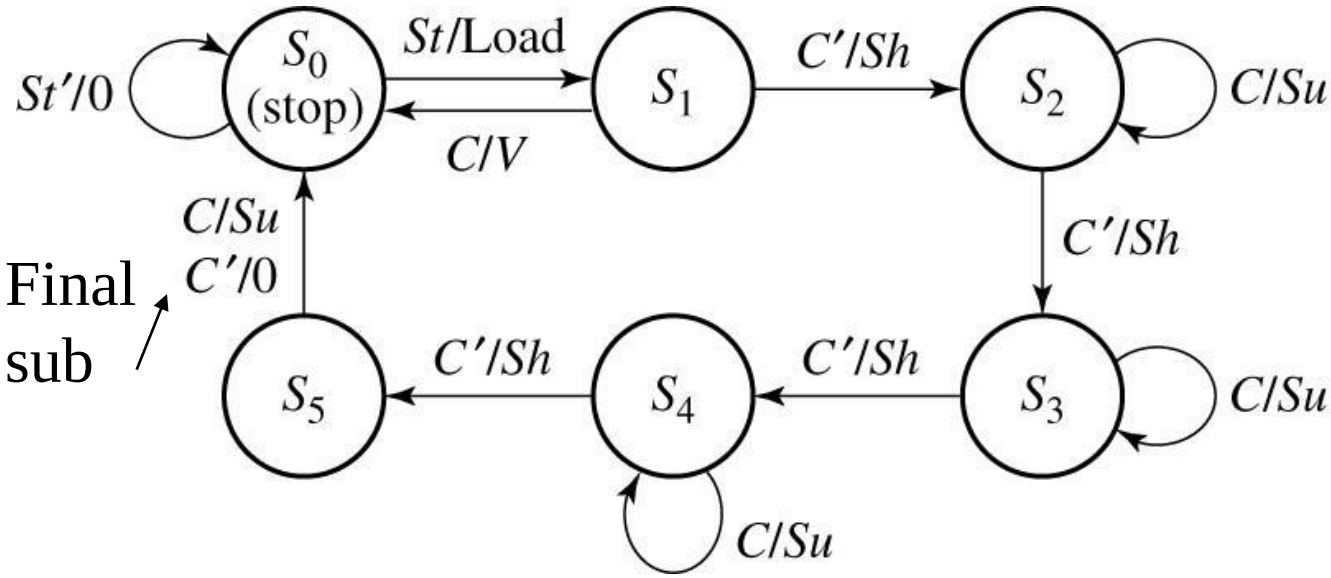
- St: start signal. St =1, load 8-bit dividend and 4-bit divisor into registers. If C = 1 at this moment, this is overflow. Set V flag.



Binary Division (cont.)

- St: start signal. St =1, load 8-bit dividend and 4-bit divisor into registers. If C = 1 at this moment, this is overflow. Set V flag.
- Normally, C =0 at first.
- After C=1 (do sub), then C will always be 0.

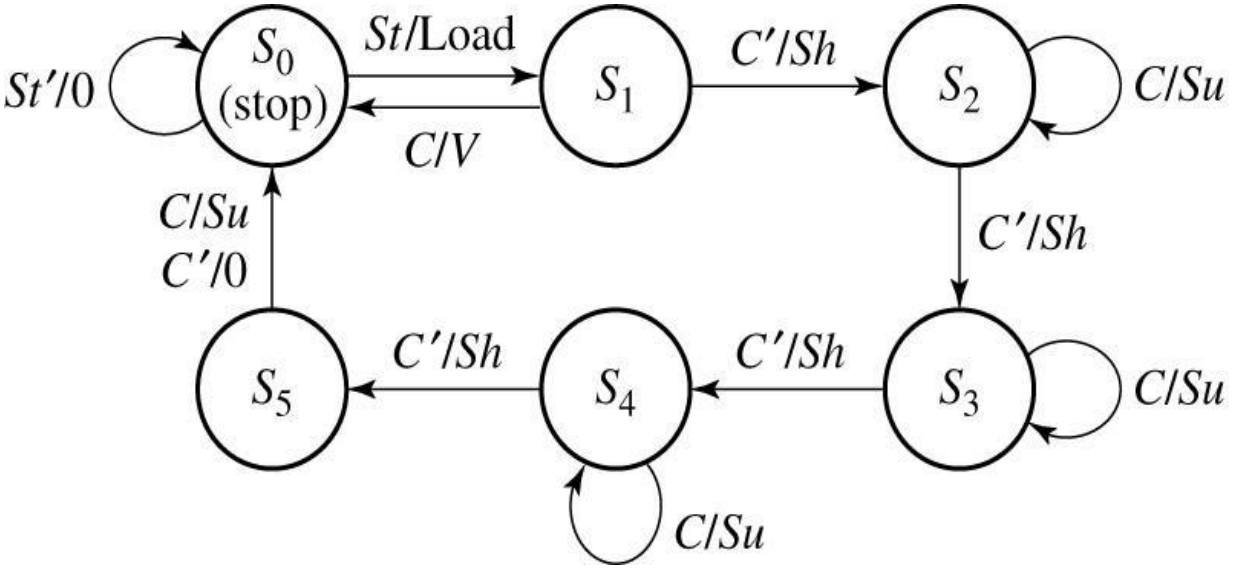




Binary Division (cont.)

- One-hot assignment

$$Q_0^+ = St'Q_0 + CQ_1 + Q_5 \qquad Q_1^+ = StQ_0$$
$$Q_2^+ = C'Q_1 + CQ_2 \qquad Q_3^+ = C'Q_2 + CQ_3$$
$$Q_4^+ = C'Q_3 + CQ_4 \qquad Q_5^+ = C'Q_4$$
$$Load = StQ_0 \qquad V = CQ_1$$
$$Sh = C'(Q_1 + Q_2 + Q_3 + Q_4) = C'(Q_0 + Q_5)'$$
$$Su = C(Q_2 + Q_3 + Q_4 + Q_5) = C(Q_0 + Q_1)'$$



Binary Division (cont.)

- Complete circuit