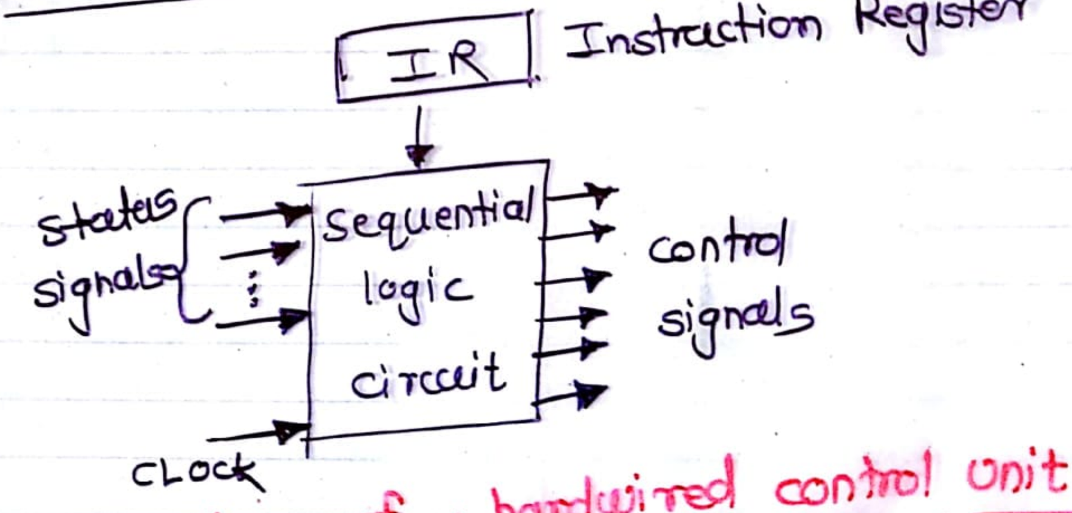
Design of control unit

- Hardwired implementation
- Micro programmed implementation.

Hardwired control unit is implemented as a sequential logic circuit or a finite state machine that generates a specific sequence of control signals to execute an instruction.

In microprogrammed control unit, microinstructions are stored in a special memory called control memory. In response to a machine instruction a set of microinstructions are executed. Each micro-instruction generates a set of control signals. Execution of micro-program (a set of micro-instruction) resembles execution of a conventional program

Hardwired control:



Structure of a hardwired control unit

Hardwired control unit uses a fixed logic to interpret an instruction and generate appropriate control signals. while designing a control unit, we have to consider various factors like.

① Amount of hardware used.
② The speed of operation.
③ cost of design.

Four techniques to design hardwired control unit

① state table method; It is a classical method of sequential circuit design. It attempts to minimize the amount of hardware.

② Delay element method: It is heuristic method based on the use of clocked delay element (D-flipflop) for control signal timings

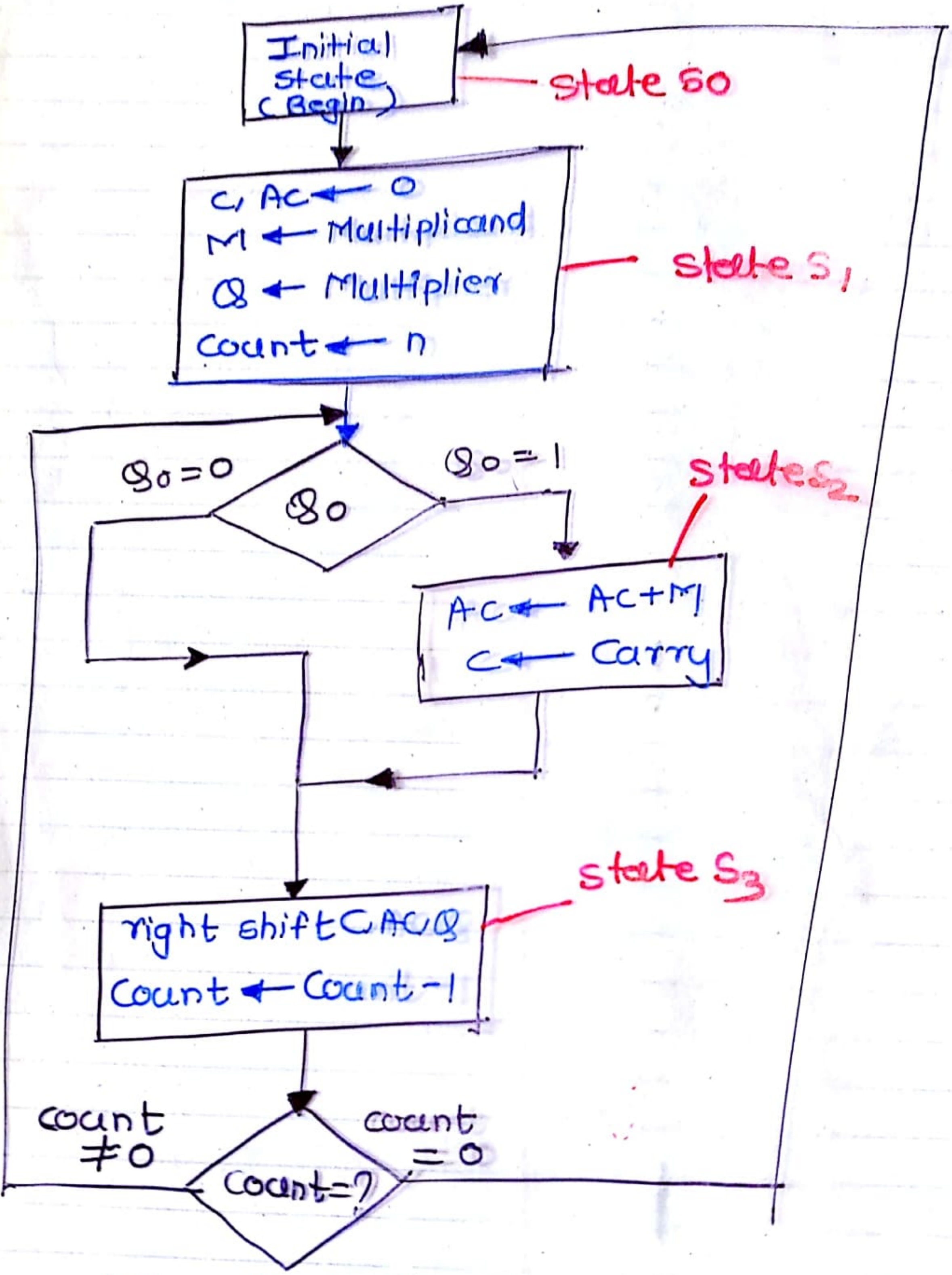
③ Sequence counter method: It uses counter for timing purposes.

④ PLA method: It uses programming logic arr

State Table method: start with the constructi of state transition table, In every state the contr unit generates a set of control signals. Control unit transmits from one state to another state depending on its;

① current state
② Input to the controller

Hardwired control unit for multiplication of two unsigned numbers



Flow chart for binary multiplication

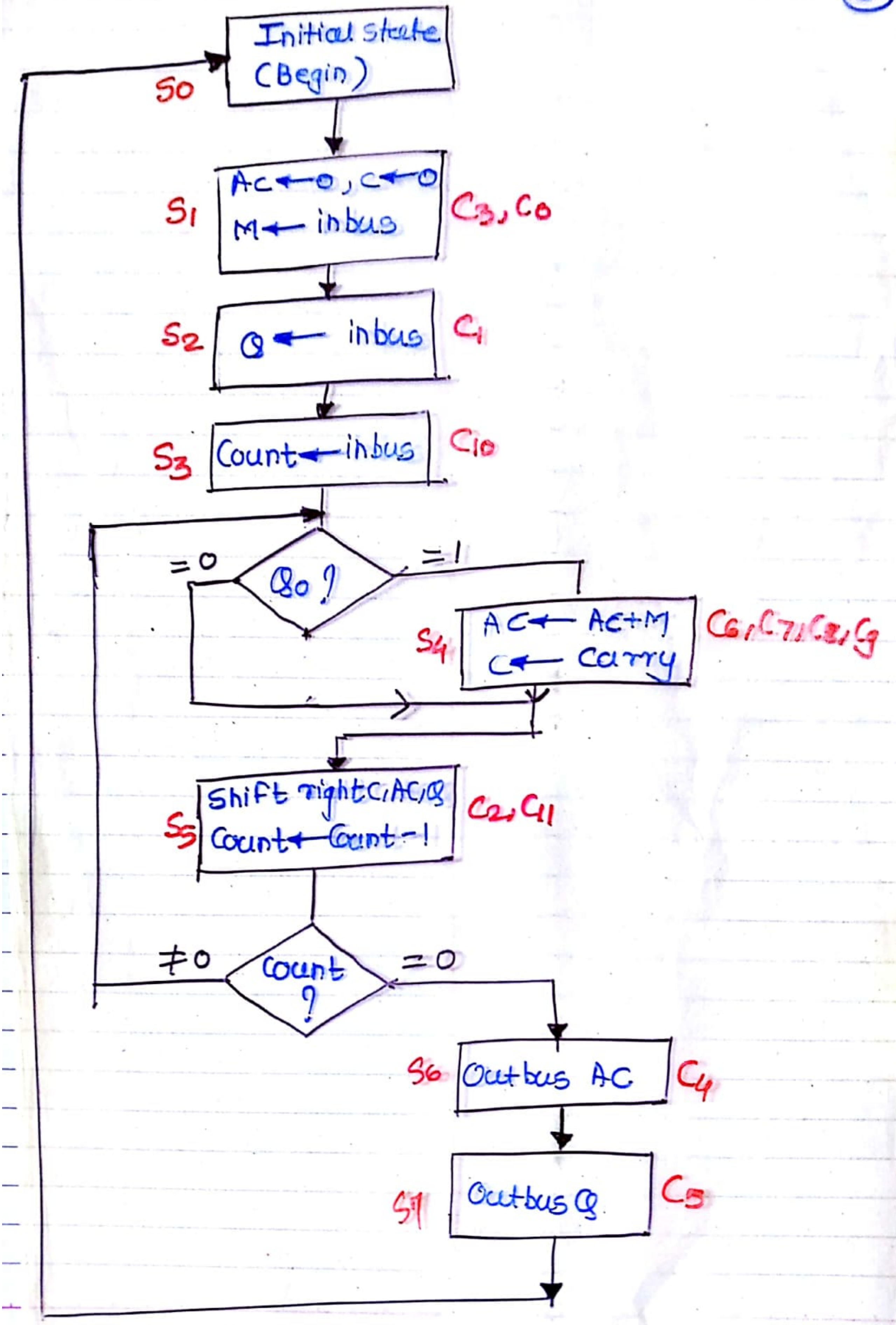
Initially load multiplicand in M and multiplier in Q register C handles carry if any during addition. Initially, registerC and AC are cleared and the sequence counter register count is set to n, which is equal to number of bits in the multiplier.

Next we enter a loop that keeps forming the partial products. The multiplier bit Q_0 is checked, and if it is equal to 1, multiplicand M is added to AC. Any carry from addition is transferred to C. The counter is decremented by 1 and register C, AC, Q are then shifted once to the right to obtain a new partial product.

Flowchart given in fig ① is redrawn in fig ②. operations mentioned in state S_1 of fig ① can not be processed in parallel and hence they have been seperated as shown in fig ②., control unit is as shown in fig ③

Control specification: In each state, the control unit generates a set of control signals required to perform operations related to the state.

- Multiplication starts after receiving the control signal 'Begin' in state 0
- In state S_1, control signal C_3 clears register AC and C. Multiplicand is loaded in register M through inbus using control signal C_0

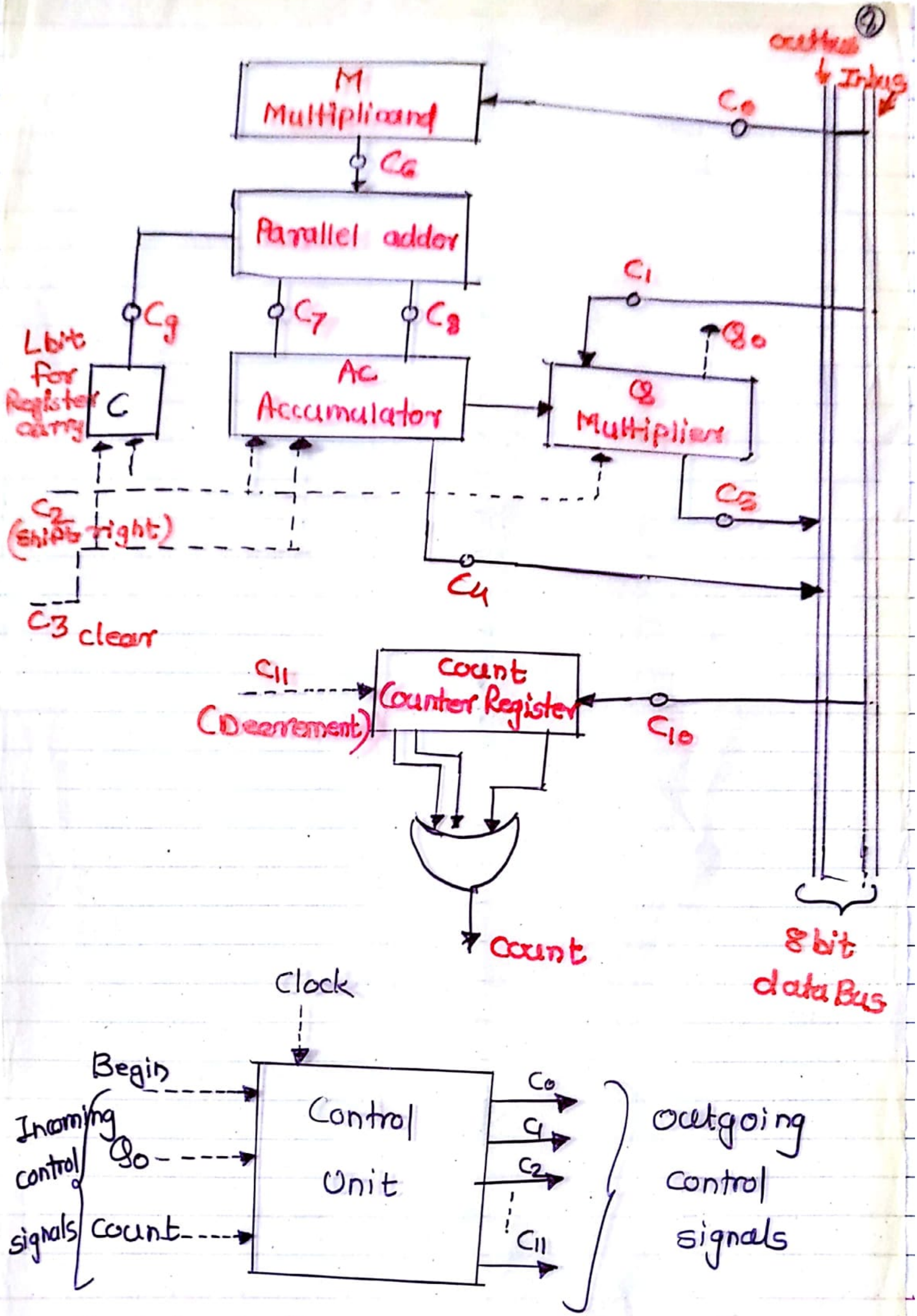


Flow chart for binary multiplication (with control signals)

- In state S_2, control signal C_1 is used to load multiplier in register Q through inbus.
- In state S_3, sequence counter register is loaded with initial count (number of bit in multiplier) through inbus. control signal C_{10} is used.
- In state S_4, operation
  $$AC \leftarrow AC+M$$
  is performed using control signals C_6, C_7, C_8, carry out is loaded in register C using the control signal C_9, Sequence counter register is decremented by 1 using control signal C_{11}
- In state S_4' counter is decremented by 1 using control signal C_{11}
- In state S_5, C, AC, Q registers are shifted right by 1 bit using the control signal C_2
- On completion of multiplication, the control unit enters state S_6 and then state S_7. Result of multiplication is sent through outbus using control signals C_4 and C_5.

Control. signals	operations.
Begin	Beginning of multiplication operation. It is activated on execution of a multiplication operation
Q_0	This is the LSB of multiplier. This determines whether add ($AC \leftarrow AC+M$) and shift will be performed or a simple shift will be sufficient

control signals	operations
Count	For a n-bit number, multiplication requires n cycles, After n-cycles signal count will become 0
C_0	Transfer multiplicand on input bus to M
C_1	Transfer multiplier on input bus to Q
C_2	Right shift C, AC, Q registers.
C_3	clear AC and C(carry) registers.
C_4	Transfer AC to output bus.
C_5	Transfer Q to output bus
C_6	Transfer M to adder for addition
C_7	Transfer AC to adder for addition
C_8	Transfer adder output to AC
C_9	Transfer carry-out from addition to C-register.
C_{10}	Transfer initial count value from input bus to counter register.
C_{11}	Decrement counter.



Control unit for multiplication of two unsigned binary numbers.

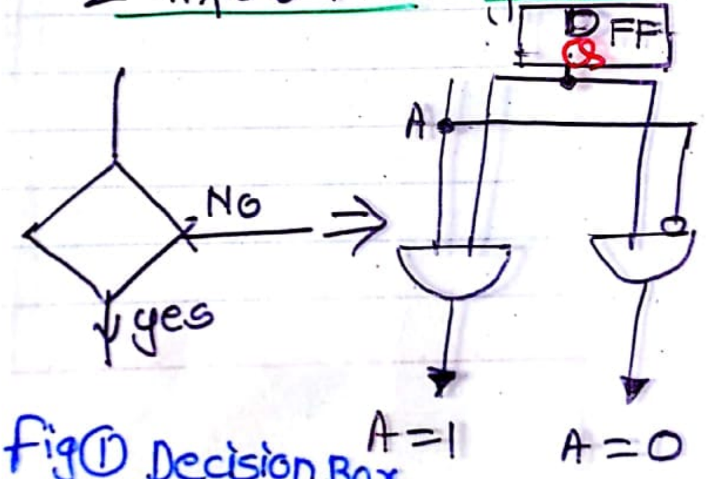
Delay-element Method

- The control signals from the control unit are activated in proper sequence.
- There is a specific time delay between activation of two groups of consecutive control signals.
- A sequence of delay element can be used to generate control signals one after the other.
- To ensure synchronous operation, the delay elements are implemented by D flip flops and controlled by a common clock signal.

Rules for delay elements:

A control unit using delay elements can be constructed directly from the flowchart that specifies required control signal sequences.

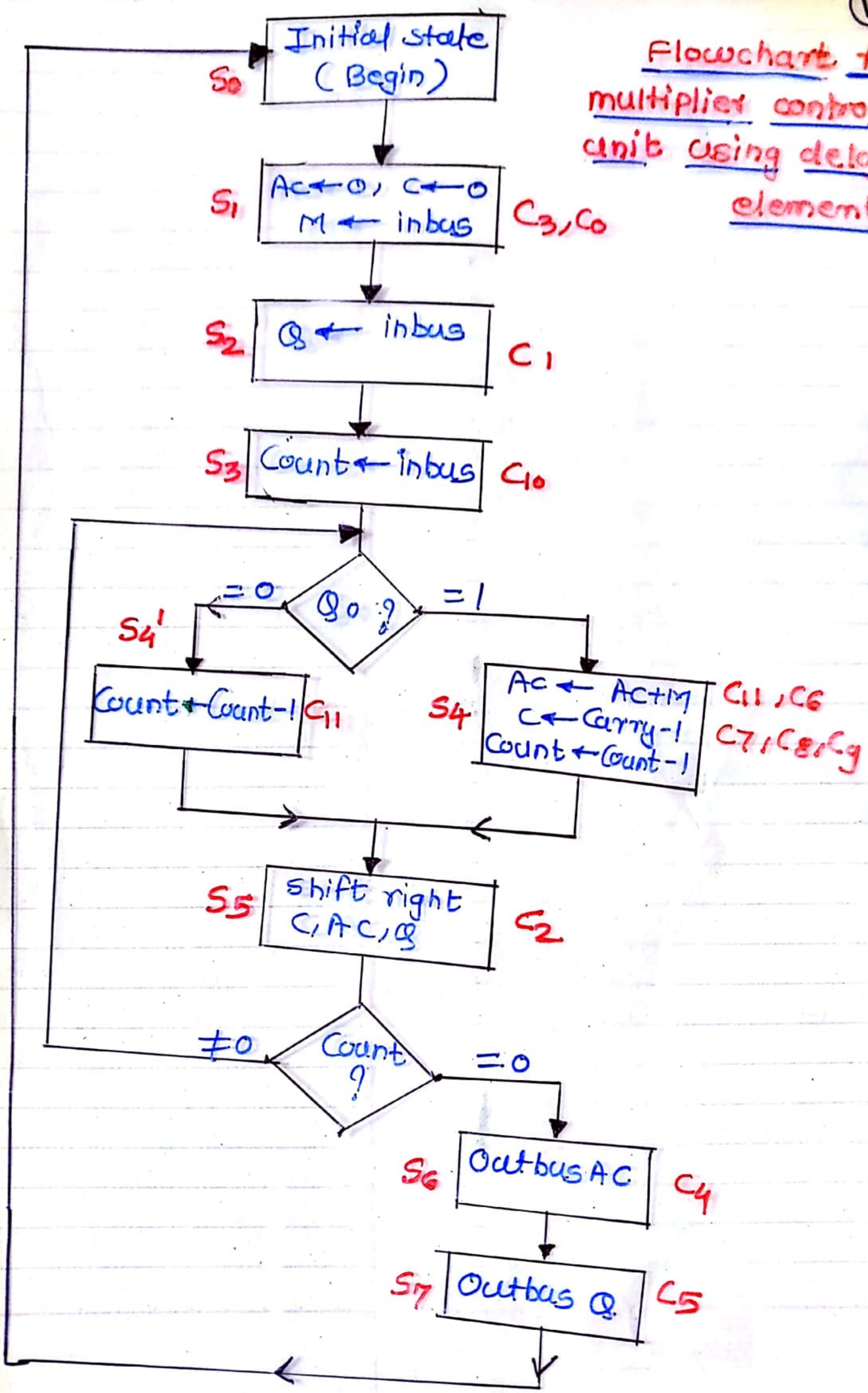
- Every state requires a delay element
- The signals that activate same control signals are ORed to get one common output signal.
- When n lines in the flowchart merge to a common point then these lines are connected to an 'n' input OR Gate.
- A decision box can be implemented by two 2-input AND gates as shown in Fig①

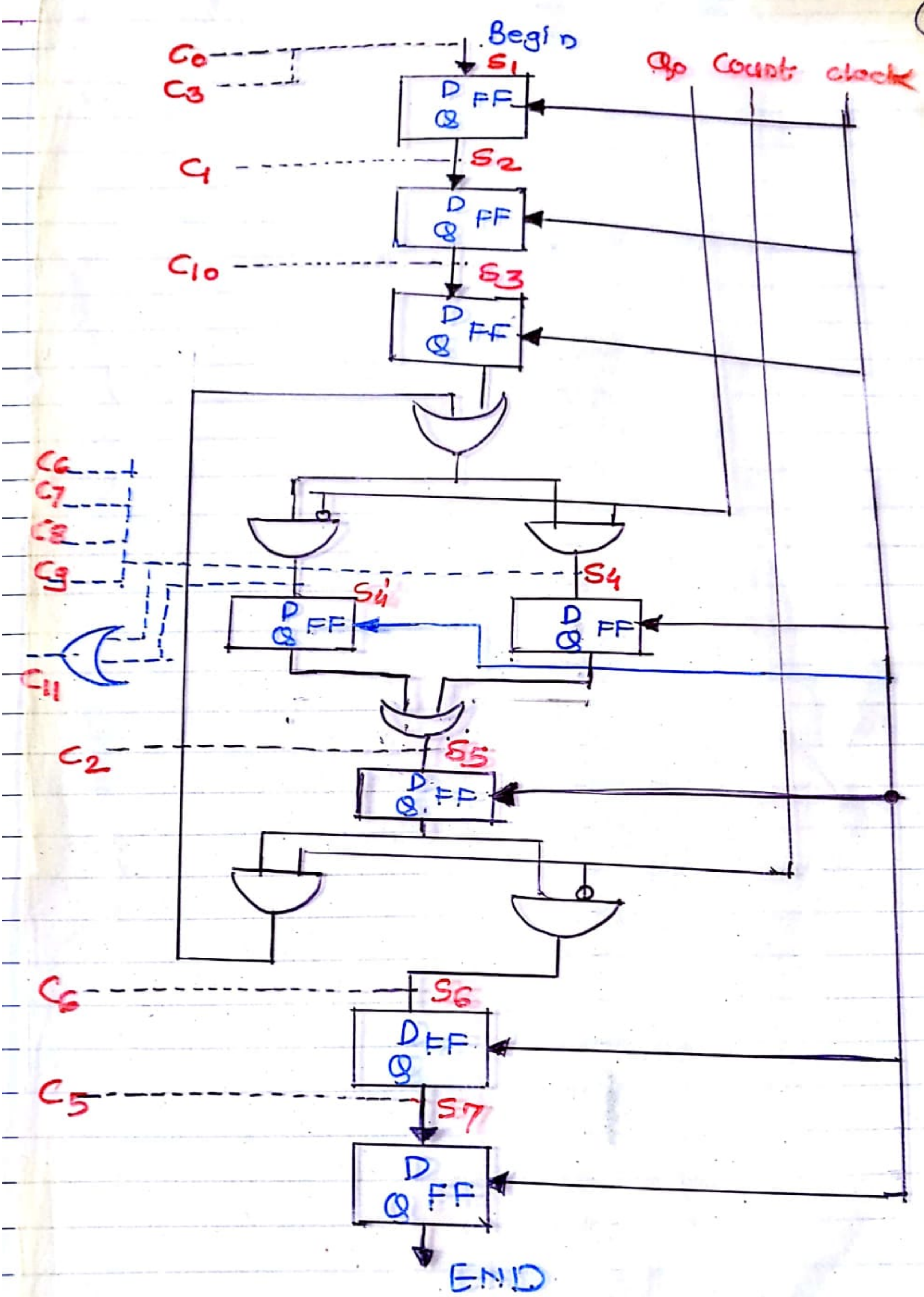


Fig① Decision Box

First I/P of each AND gate is driven by input A and complement of A respectively, while the second input of both gates is common and it is the o/p of Delay element.

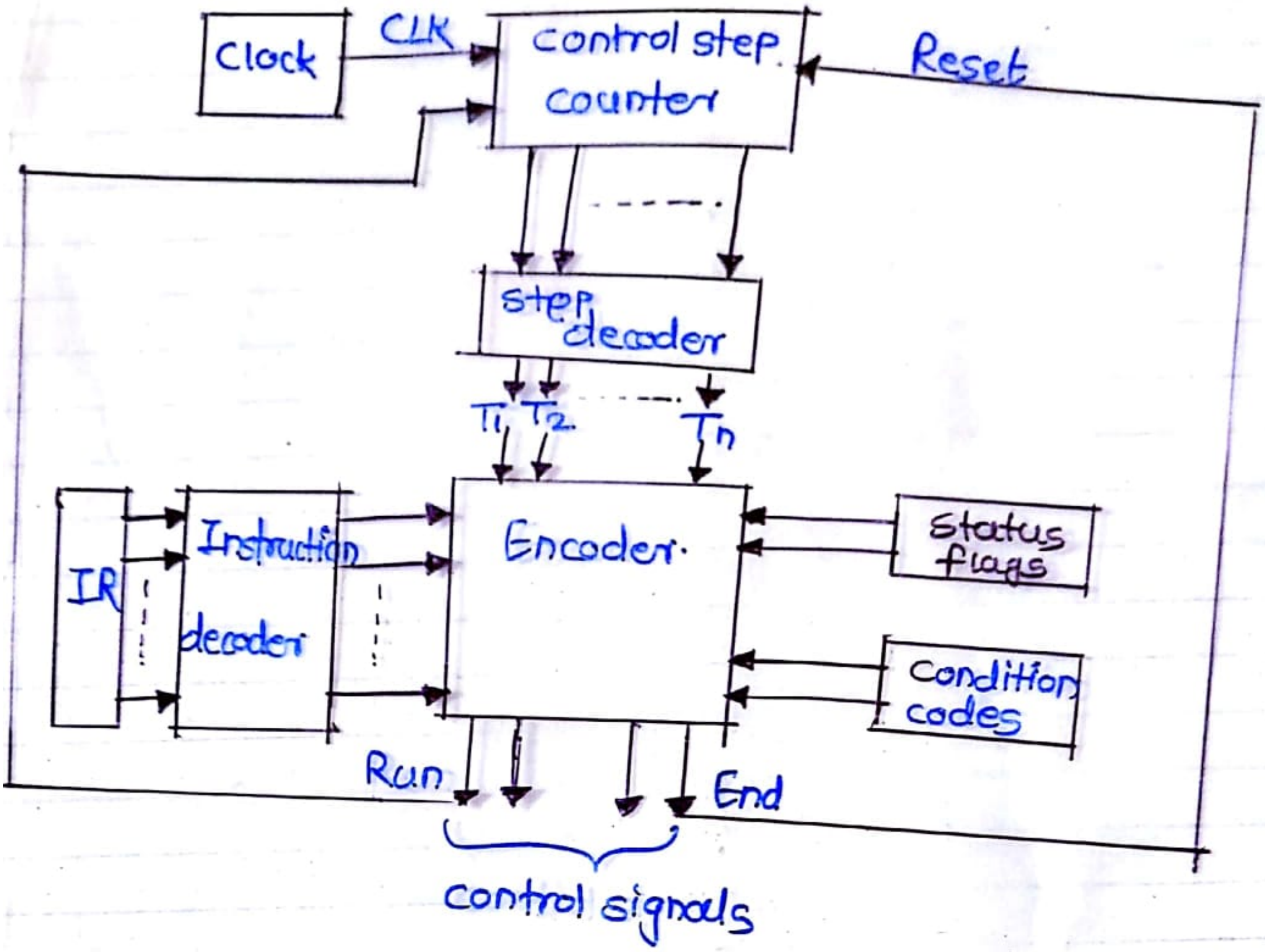
Flowchart for multiplier control unit using delay elements





Multiplier control unit using delay elements

Block Diagram of a hardwired control unit



Control signals are generated in a sequence. In each time slot, control signals are generated to perform the operation specified in the corresponding step. All time slots are kept of equal duration. The control unit may be based on the use of a counter driven by a clock signal CLK. control signals are uniquely determined by the following information:

1) Contents of the control step counter
2) Contents of the instruction register
3) Contents of the condition code and other status flags.

- The instruction loaded in IR is decoded by the instruction decodor.
- If IR is 8 bit then Instruction decoder generates 2^k ie 256 lines, one for each instruction
- Depending on the code in IR, only one line amongest all output lines of decoder is set to 1.
- step decodor provides a seperate signal line for each step or time slot in a control sequence. All the time slots are kept of equal duration.
- Instruction decoder, step decodor, status flags, condition codes provide input to encodor.
- Encodor uses these inputs to generate individual control signal.
- After execution of each instruction it generates END signal.
- This signal resets the control step counter and makes it ready for generation of control step for neaet instruction.