SCRIBE 221(DIGITAL DESIGN)
BABUL DOLEY-120101013
DATE-07/11/2013
BEDANTA BASUMATARY-120101015

AVESH MEENA-120101011

SUQUENTIAL BINARY MULTIFLICATION

Introduction:

A hardware algorithm for binary multiflication propose the register configuration for its implementation and ASMD chart to design datapath and its controller.

*This system will examine multiplies two unsigned binary numbers

*Its is a efficient that use only one adder and a shift register but it takes multiple clock cycles to form its result.

REGISTER CONFIGURATION

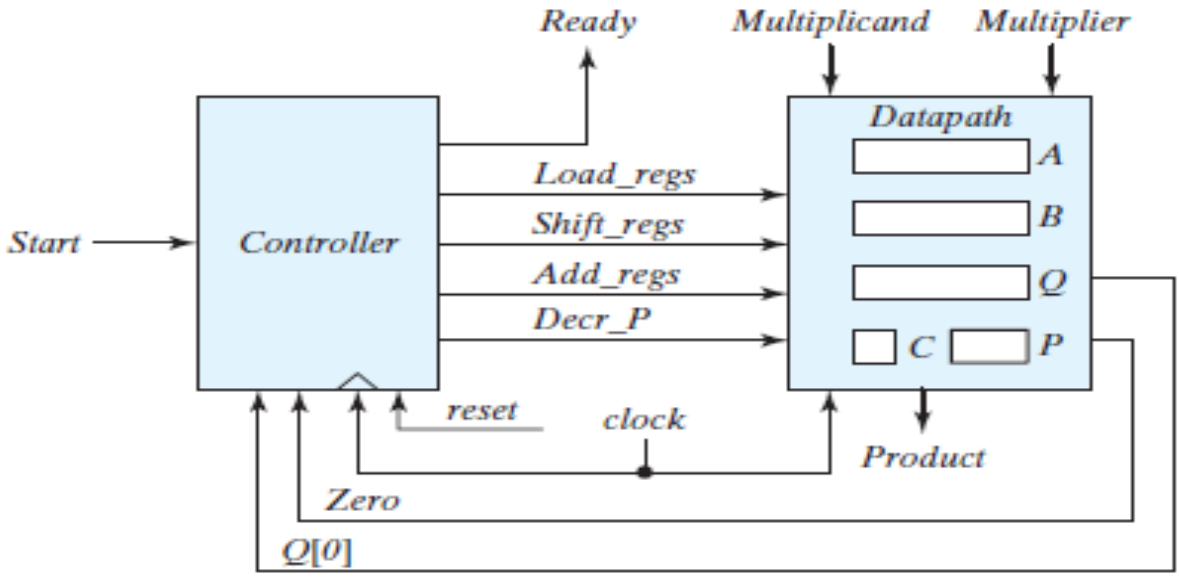
The multiplicand is stored in register B, the multiplier is stored in register Q, and the partial product is formed in register A and stored in A and Q.

* A parallel adder adds the contents of register B to register A. The C flip-flop stores the carry after the addition.

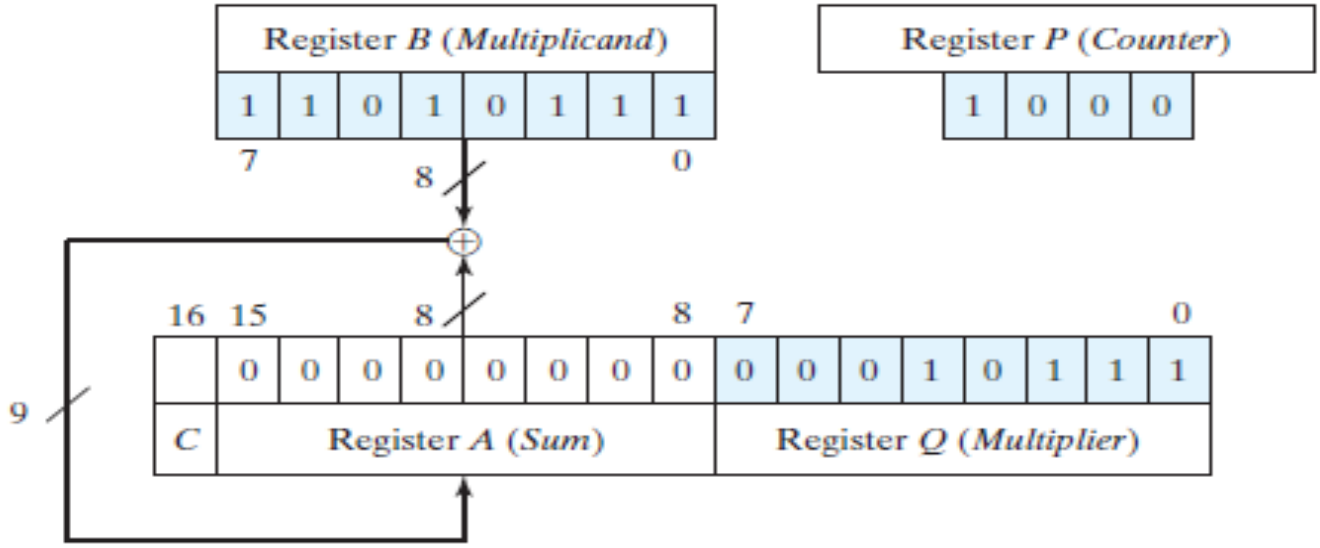
*The counter P is initially set to hold a binary number equal to the number of bits in the multiplier.This counter is decremented after the formation of each partial product.

* When the content of the counter reaches zero, the product is formed in the double register A and Q, and the process stops. The control logic stays in an initial state until Start becomes 1.

*The system then performs the multiplication. The sum of A and B forms the nmost significant bits of the partial product, which is transferred to A .



(a)



* The output carry from the addition, whether 0 or 1, is transferred to C. Both the partial product in A and the multiplier in Q are shifted to the right. The least significant bit of A is shifted into the most significant position of Q , the carry from C is shifted into the most significant position of A , and 0 is shifted into C. *After the shift-right operation, one bit of the

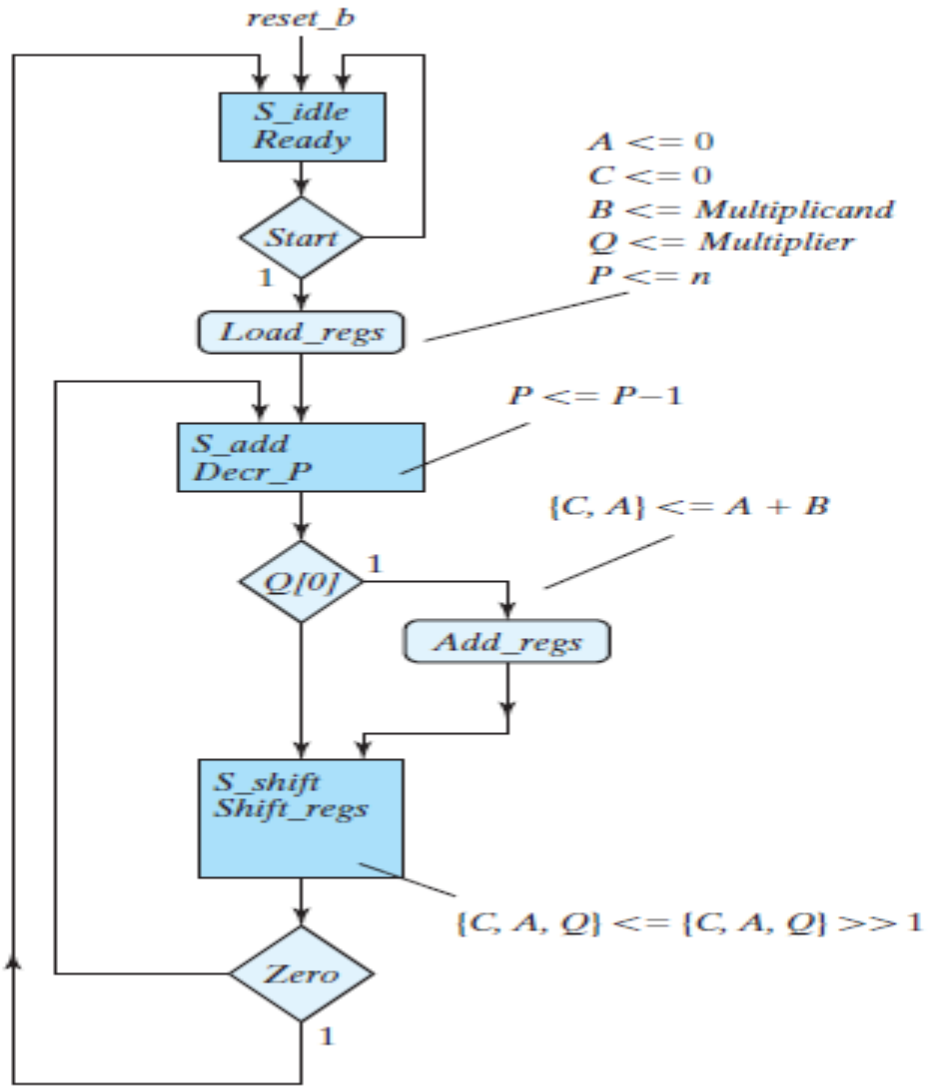
partial product is transferred into Q while the multiplier bits in Q are shifted one position to the right.

*The least significant bit of register Q , desig- nated by Q[0] , holds the bit of the multiplier that must be inspected next. The control logic determines whether to add or not on the basis of this input bit. The control logic also receives a signal, Zero, from a circuit that checks counter P for zero. Q[0] and Zero are status inputs for the control unit.

*The input signal Start is an external control input. The outputs of the control logic launch the required operations in the registers of the datapath unit.

*Signal Load_regs loads the internal registers of the datapath, Shift_regs causes the shift register to shift, Add_regs forms the sum of the multiplicand and register A , and Decr_P decrements the counter.

ASMD chart



*Initially, the multiplicand is in B and the multiplier in Q. As long as the circuit is in the initial state and Start=0, no action occurs and the system remains in state S_idle with Ready asserted.

*The multiplication process is launched when Start=1. Then, 1 con-trol goes to state S_add, 2 register A and carry flip-flop C are cleared to 0, 3 registers B and Q are loaded with the

multiplicand and the multiplier, respectively, and 4 the sequence counter P is set to a binary number n, equal to the number
of bits in the multiplier.

*In state S_add , the multiplier bit in Q[0] is checked, and if it is equal to 1, the multiplicand in B is added to the partial product in A . The carry from the addition is transferred to C .

*The partial product in A and C is left unchanged if Q[0] = 0. The coun-ter P is decremented by 1 regardless of the value of Q[0] , so Decr_P is formed in state S_add as a Moore output of the controller.

*The next state is S_shift . Reg-isters C, A , and Q are combined into one composite register CAQ , denoted by the concatenation {C, A, Q}, and its contents are shifted once to the right to obtain a newpartial product.

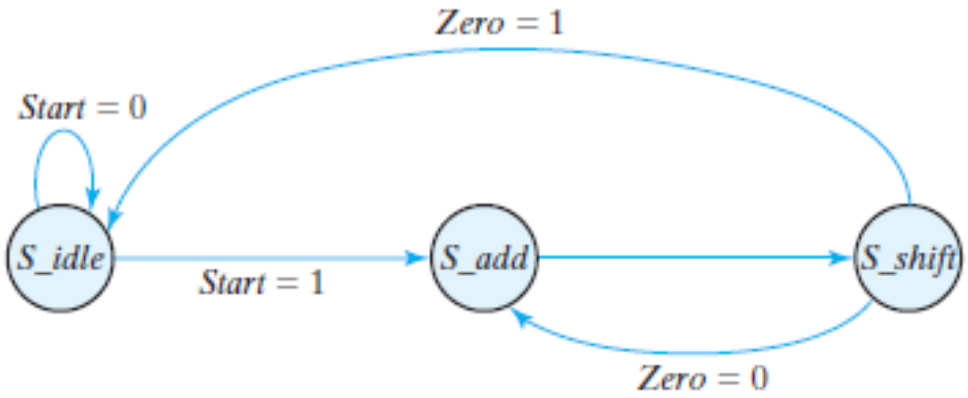
*The shift operation can be described by the following register operations:

$A \leftarrow$-shr A, $A_{n-1} \leftarrow$- C

$Q \leftarrow$-shr Q, $Q_{n-1} \leftarrow -A_o$

$C \leftarrow 0$

CONTROL LOGIC



State Transition		Register Operations
From	To	
S_idle		Initial state
S_idle	*S_add*	$A <= 0, C <= 0, P <= dp\_width$
S_add	*S_shift*	$P <= P - 1$ if $(Q[0])$ then $(A <= A + B, C <= C_{out})$
S_shift		shift right $\{CAQ\}, C <= 0$

For the multiplier, these signals are Load_regs (for parallel loading the registers in the datapath unit),Decr_P (for decrementing the counter), Add_regs (for adding the multiplicand and the partial product), and Shift_regs (for shifting register CAQ).

* The inputs to the controller are Start, Q[0] , and Zero, andthe outputs are Ready, Load_regs, Decr_P, Add_regs , and Shift_regs , as specified in the ASMD chart.
*Q[0] affects only the output of the controller, not its state transitions. The machine transitions from S_add to S_shift unconditionally.

REGISTER AND DECODER

The sequence-register-and-decoder uses a register for the control states and a decoder to provide an output corresponding to each of the states .
A binary multiplier have three states and two inputs. To design with a sequence and decoder, we need two flip flops for the register and two-to-four line decoder.

State Table for Control Circuit

	Present State		Inputs			Next State					
Present-State Symbol	G_1	G_0	*Start*	*Q[0]*	*Zero*	G_1	G_0	*Ready*	*Load_regs*	*Decr_P*	*Add_regs*
S_idle	0	0	0	X	X	0	0	1	0	0	0
S_idle	0	0	1	X	X	0	1	1	1	0	0
S_add	0	1	X	0	X	1	0	0	0	1	0
S_add	0	1	X	1	X	1	0	0	0	1	1
S_shift	1	0	X	X	0	0	1	0	0	0	0
S_shift	1	0	X	X	1	0	0	0	0	0	0

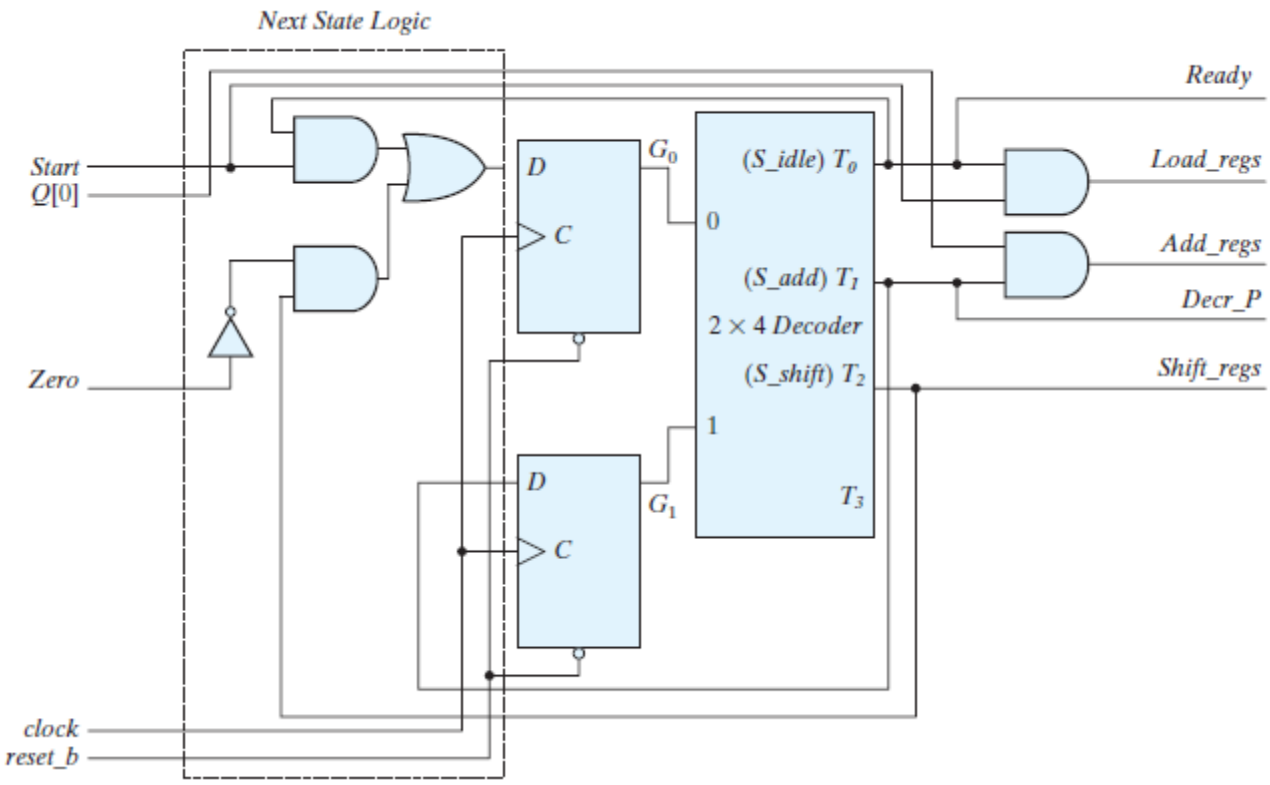
Conditions can be specified by the equation :

$$D_{G1}=T_1$$

where D_{G1} is the D input of flip flop G1.

The D input of Go is

$$D_{G0}=T_0.Start+T_2.Zero'$$



Logic diagram of control using sequence and decoder

It consists of a registers with two flip-flops G_1 and G_0 and a 2 * 4 decoder.

The outputs of the decoder are used to generate the inputs to the next-state logic as well as the control outputs. The outputs of the controller should be connected to the datapath to activate the required register operations.