EEE336 June 15 Solutions

1. a. Cache memory is Random Access Memory (RAM) which is used as a temporary store for data and instructions. It is faster than main memory as it is located on the CPU chip itself or on a separate chip with a high speed interconnect to the CPU. **(2)**

Direct Mapping: Each block in main memory is mapped to a single fixed location in the cache memory. **(2)**

Fully Associative: A block in main memory may be mapped to any cache location. **(2)**

Set Associative: The cache memory is divided into a number of SETS. A block in main memory is mapped to a single unique SET but within the SET, the mapping of the block is associative. **(2)**

b.

N	Dividend, D	2^n x V	D - 2^n x V	Quotient bit
3	00001111	00110000	11011111	0
2	00001111	00011000	11110111	0
1	00001111	00001100	00000011	1
0	00000011	00000110	11111101	0

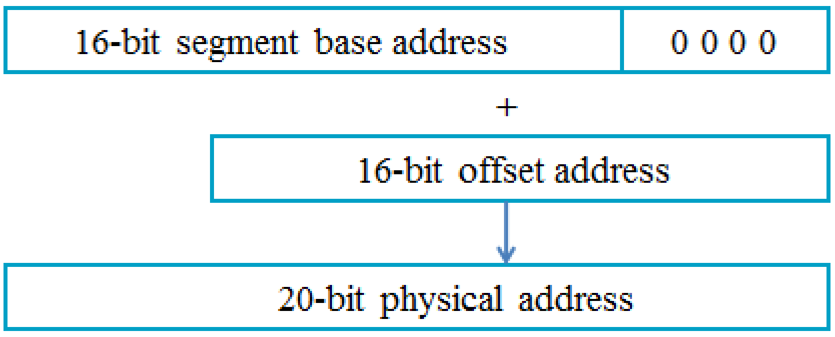
Quotient = 00000010 (2)

Remainder = 00000011 (3) **(6)**

c. 1Mb of memory requires 20 address bits.

The 8086 has four 16-bit segment registers which can each address a 64K block of memory. To form an address, a segment address is combined with an offset address.

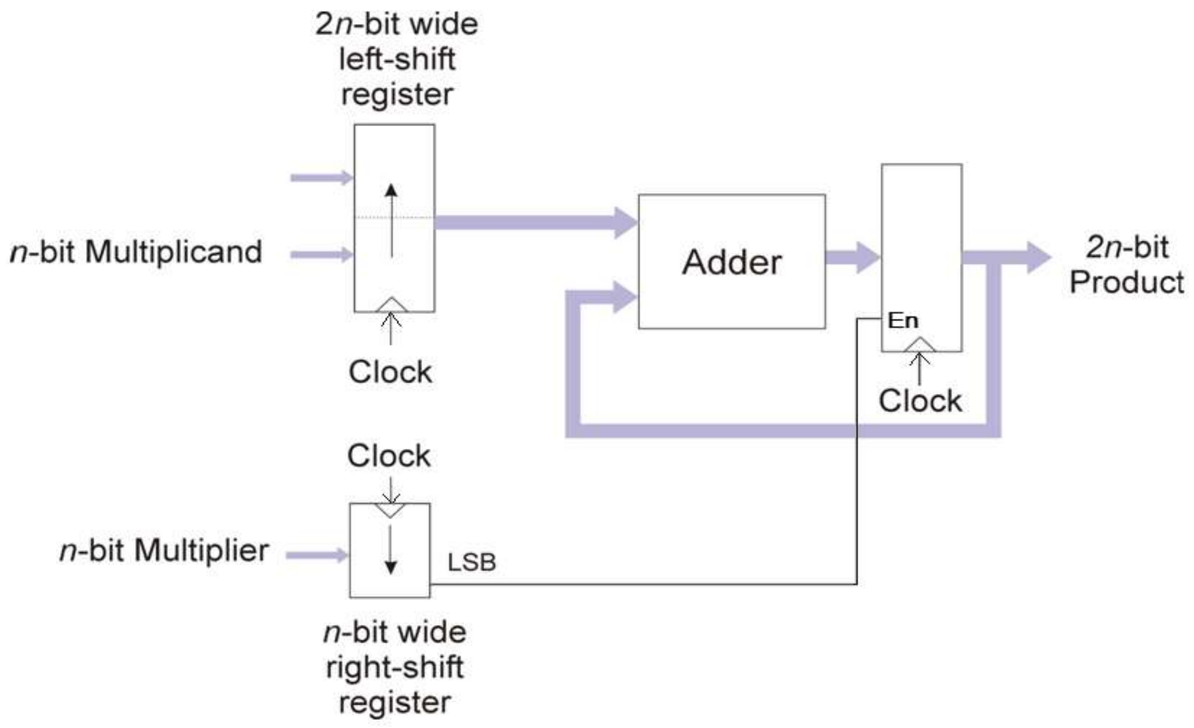
The segment address is shifted four bits to the left and added to the offset address.



(6)

2. **a.**

Assuming *n*-bit wide multiplicand and multiplier, a possible circuit diagram is:



i) The multiplicand is loaded into the bottom *n* bits of the 2*n*-bit wide multiplicand shift register. (The upper *n* bits are zero.)

ii) The multiplier is loaded into the multiplier shift register.

iii) The 2n-bit wide accumulator register is loaded with zero. (This register holds the running total of the partial products.)

The purpose of the multiplier shift register is to extract the *m*th bit of the multiplier; simultaneously, the multiplicand shift register will output a version of the multiplicand left-shifted by *m* places which comprises the potential partial product. Depending on the value of the *m*th multiplier bit, the would-be partial product is either added to the running total of partial products (if the *m*th multiplier bit is 1) or ignored (if the *m*th multiplier bit is 0).

After *n* clock cycles, the 2*n*-bit wide product register holds the product. **(6)**

b.

Time	Contents of 2*n*-bit wide multiplicand shift register	Multiplier shift register	Output of adder	Contents of 2*n*-bit wide accumulator register
1.	00001010	0110	00001010	00000000
2.	00010100	*011	00010100	00000000
3.	00101000	**01	00111100	00010100
4.	01010000	***0	10001100	00111100
Final	Don’t care	****	Don’t care	00111100

(5)

c. An unsigned n-bit number can be represented by:

$$a_{n\text{-}1}2^{n\text{-}1} + \ldots + a_2 2^2 + a_1 2^1 + a_0 2^0$$

whereas a 2s-complement number can be represented by:

$$-a_{n\text{-}1}2^{n\text{-}1} + \ldots + a_2 2^2 + a_1 2^1 + a_0 2^0$$

The process of shift-and-add multiplication involves left-shifting the multiplicand and (conditionally) adding this to form a partial product. The problem with 2s-complement numbers comes from the negative sign of the $a_{n\text{-}1}$ multiplicand digit. When this is left-shifted, say one place, to give a number of the form:

$$-a_{n\text{-}1}2^{n} + \ldots + a_2 2^3 + a_1 2^2 + a_0 2^1$$

the weighting of the digit in the nth column is incorrect (i.e. it is negative). This difficulty is repeated for all left shifts of the multiplicand with the result that all the partial products are incorrect and therefore the product itself is meaningless.

(6)

d. As n becomes moderate to large, the speed of operation of the circuit will be dictated by the propagation delay through the adder.

(3)

3.a.

A stack is generally implemented in a special reserved block of main memory, typically at the very top of memory. By convention, stacks grow down in memory. The processor maintains the address of the top of the stack in a special Stack Pointer (SP) register; the contents of SP point to the datum at the top of the stack.

Two operations are defined on a stack: PUSH and POP. The PUSH operation decrements the value in SP and then transfers a value to the address pointed to by SP. The POP operation copies the datum pointed to by SP and then increments the value in SP. In this way, the processor is able to store data in a last-in-first-out manner.

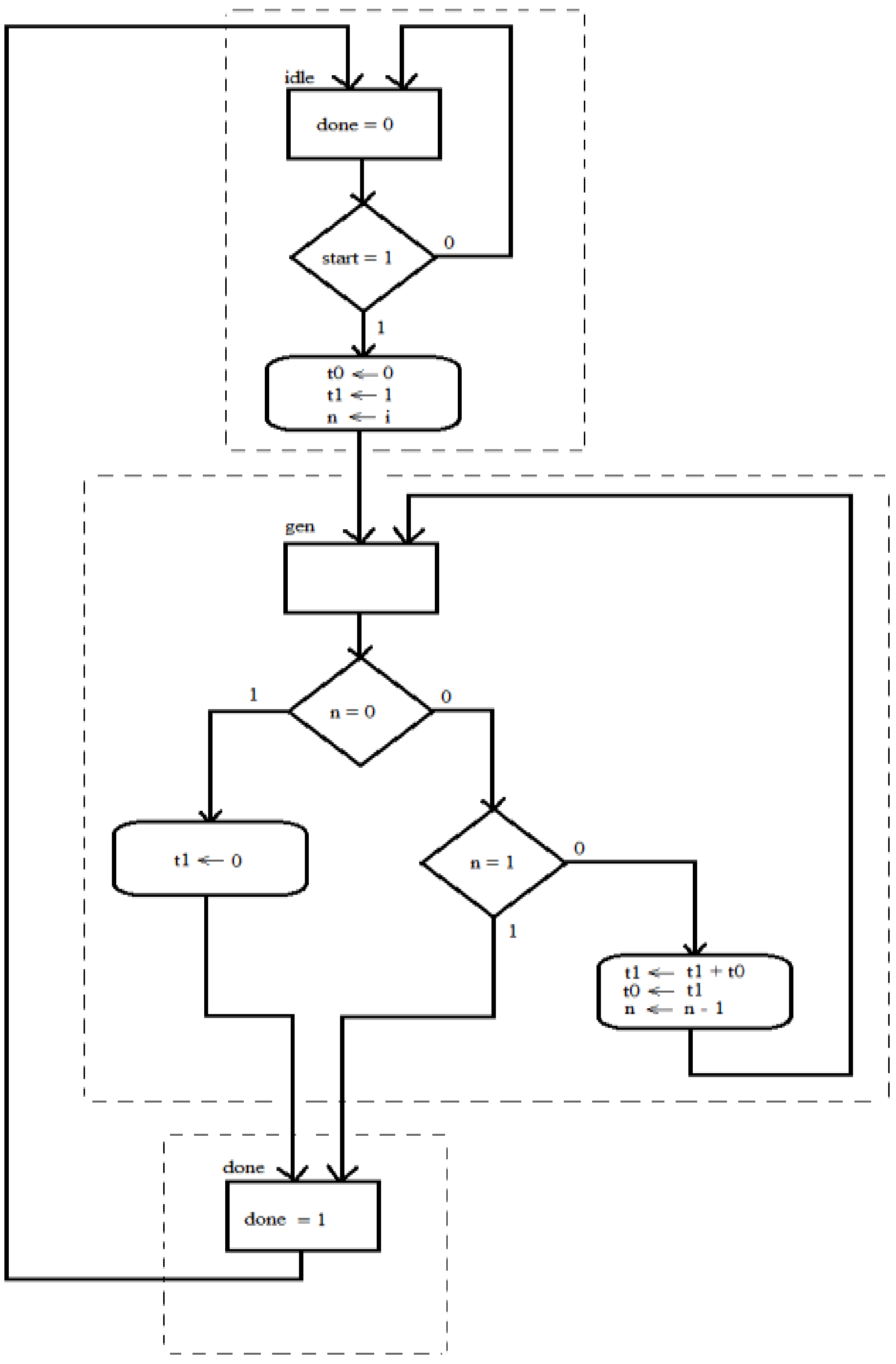
(4)

b.

Typically, a procedure would have a list of parameters. When a procedure is called, the current state of the CPU (register values, instruction pointer, etc.) is saved by being pushed onto the stack (from whence it can be restored by popping off the stack). Similarly, parameters are pushed onto the stack from where they can be accessed by addressing relative to the current SP value.

(4)

c.



(8)

IDLE: This is the reset state and the state machine will stay in this state until it receives a **start** signal. The state machine will return to this state after generation of a complete sequence setting the **done** flag to zero. Two temporary registers used to compute the value for when **i** > 1 are initialized and the number of values to be generated in the sequence is read from the input **i** into a counter n at the next clock edge.

GEN: This state where the sequence is generated. The count **n** is decremented each time round the loop and the registers t1 and 10 are used to perform the calculation. A special case exists for when **i** = 0.

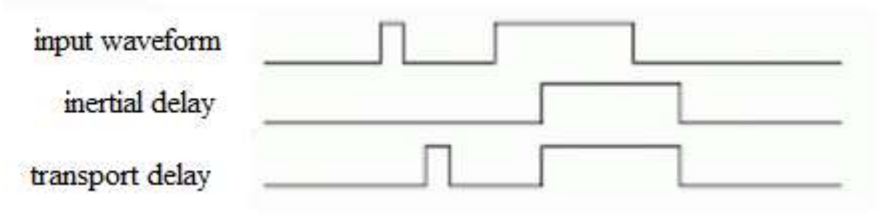
DONE: This state is used to indicate the end of the sequence by going high for 1 clock cycle.

(4)

4. a.

Inertial Delay is a delay model which indicates the amount of time that a signal change must persist at the input of a logic gate in order for a change to propagate to the output. A pulse which does not last longer than the inertial delay will be suppressed and there will not be any change at the output. Pulses of length greater than the inertial delay will propagate to the output after that delay.

Transport delay does not have this restriction and changes on the input will be reflected at the output after the specified delay.



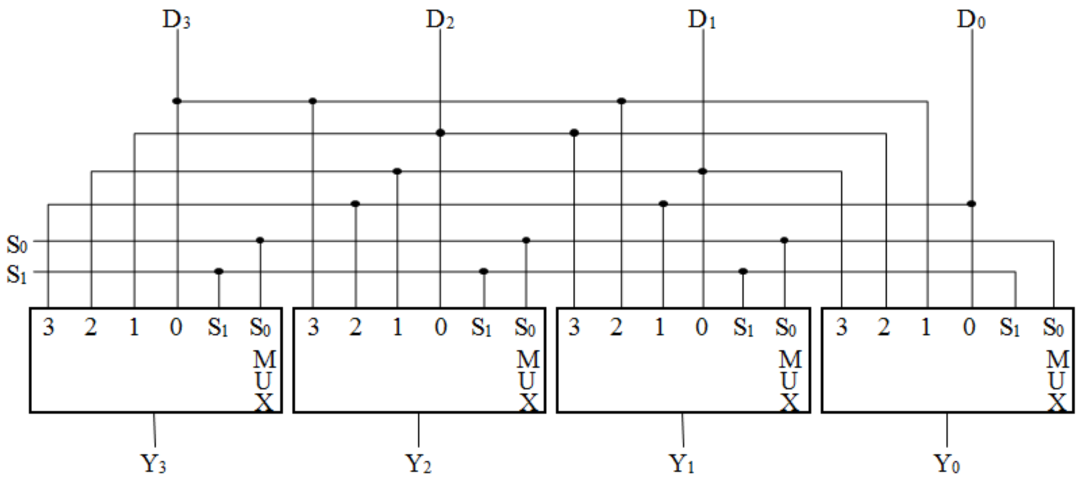
(6)

b.

a1	a0	y0	y1	y2	y3
0	0	1	0	0	0
1	0	0	1	0	0
0	1	0	0	1	0
1	1	0	0	0	1

2-to-4 address decoder **(4)**

c. A barrel shifter performs a rotate in which the bits shifted out are inserted into the positions vacated.



(6)

d.

With a combinatorial shifter, any shift amount can be carried out in a single clock cycle by storing the data in appropriate registers between the shifter.

With a flip-flop based solution, the shifter must first be loaded and then n clock cycles applied for an n place shift.

(4)