EE3-05

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2012

EEE/ISE PART III/IV: MEng, BEng and ACGI

Corrected Copy

DIGITAL SYSTEM DESIGN

Tuesday, 8 May 2:30 pm

Time allowed: 3:00 hours

There are FOUR questions on this paper.

Answer ALL questions.

All questions carry equal marks.

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible	First Marker(s) :	C. Bouganis
	Second Marker(s) :	T.J.W. Clarke

Special information for invigilators:
None

Information for candidates:
In the figures showing digital circuits, all components have, unless explicitly indicated otherwise, been drawn with their inputs on the left and their outputs on the right. All signals labelled with the same name are connected together. All circuits use positive logic. The least significant bit of a bus signal is labelled as bit 0, and the most significant bit with the highest integer number. Therefore the signal X[7:0] is an eight bit bus with X7 being the MSB and X0 the LSB.

Hexadecimal numbers are prefixed with $. For example the decimal number 10 is written as $A.

In questions involving circuit design, you may use any standard digital circuits that are not explicitly forbidden by the question provided that you fully specify their operation.

Marks may be deducted for unnecessarily complex designs unless you are explicitly instructed not to simplify your solution.

The Questions

1. The CORDIC algorithm is based on the idea of rotating a vector $\mathbf{v} = (x, y)$ by an angle θ using a specific set of elementary angles a^i, where $i \in [0, N-1]$, for a predefined set of iterations N. Starting from the basic rotation equations:

$$x^{i+1} = x^i \cos(a^i) - y^i \sin(a^i)$$

$$y^{i+1} = y^i \cos(a^i) + x^i \sin(a^i)$$

and the fact that $z^{i+1} = z^i - a^i$, where z is the angle accumulator, the hardware friendly equations for each iteration of the CORDIC algorithm are given below:

$$x^{i+1} = x^i - d_i y^i 2^{-i}$$

$$y^{i+1} = y^i + d_i x^i 2^{-i}$$

$$z^{i+1} = z^i - d_i a^i$$

where $d \in \{-1, 1\}$ is determined by some criterion, and a^i denotes the i^{th} elementary angle.

a) Draw the circuit that implements the bit-parallel CORDIC algorithm that achieves maximum throughput. The available set of elementary angles is given in Figure 1.1. State the achieved throughput of the system in results per cycle and the latency of the system in cycles.

[10]

b) Show how the CORDIC algorithm can evaluate arctan(y). Assume that the value of $K = \prod_{i=0}^{6}(1 + \tan^2 a^i)^{1/2}$ is known.

[5]

c) Draw the circuit that implements an iterative bit-parallel CORDIC algorithm when only a single add/sub module is available. You can assume that you do not have any restrictions on the number of the other supported modules. The available set of elementary angles is given in Figure 1.1. Provide the throughput of the design in results per cycle.

(Note: If your design requires the use of a Finite State Machine (FSM), you do not have to provide any details of its internal design)

[5]

a	atan(a) in degrees
1	45.000
0.5	26.565
0.25	14.036
0.125	7.125
0.0625	3.576
0.03125	1.790
0.015625	0.895

Figure 1.1 Elementary angles and their tangents

2. a) Design a general parallel multiplier that can multiply together two n-bit operands based on full-adder (FA) components and AND gates. Provide a description of its operation. The multiplier's inputs are under fixed-point number representation. Comment on how the required area scales with the number of bits n of the operands. Assume that 1 FA consumes 1 unit of area. You can ignore the area consumed by the AND gates.

[7]

b) Design a n-bit ripple-carry adder based on full-adders, and comment on how its area scales with the number of bits n of the operands. Assume that 1 FA consumes 1 unit of area.

[3]

c) For both designs (i.e. parallel multiplier and ripple-carry adder), derive the critical paths. Assuming a 1 unit delay when the signals pass through a full-adder (FA), derive an expression for the delay of critical paths as a function of the number of bits n of the operands.

[5]

d) Assuming that 1 FA consumes 1 unit of area and ignoring the area required by a NOT gate, provide a bit-parallel architecture for the following constant coefficient multiplier $14 \times x$ that requires the minimum possible area. The signal x takes integer non negative values. Derive an expression on how the required area scales with the number of bits n of the input x and compare it to the previous general parallel multiplier.

Assuming a 1 unit delay when the signals pass through a full-adder (FA), derive an expression for the delay of the critical path as a function of the number of bits n of the operand x. If your design employs NOT gates, you can safely ignore them.

[5]

3. Figure 3.1 depicts the state diagram of a 3-state finite-state machine (FSM), with an input signal A, and an output signal B. The input signal A is not synchronous with the system. When A is '1', the FSM moves to the next state, otherwise it returns to state S_0. When the FSM is in state S_2, it always returns to state S_0.

a) Ignoring metastability and using the minimum number of D flip-flops and logic, design a circuit that implements the FSM. Provide the boolean equations for the next state signals and the B output.

[7]

b) Assuming that the logic is mapped to 4-LUTs (Look Up Tables) when the target device is an FPGA, estimate and justify the number of 4-LUTs required to implement the FSM. Show the contents of all used LUTs and from which signals are driven.

[6]

c) Estimate the maximum frequency of your design. The propagation delays of the components of the system are given in Figure 3.2. Assume that the wires do not exhibit any delay, and all registers are clocked with the same clock signal. You can ignore the path from the input A.

[3]

d) A register is added to input A in order to synchronise the signal with the rest of the design. Derive the maximum rate of asynchronous transitions per second for the input signal A, such that the system can achieve a Mean Time Between Failure (MTBF) of 3600 second when the design is clocked at its maximum frequency. Assume that the metastable decay constant of the flip-flops is $\tau = 1ns$ and the metastability aperture is $T_0 = 9.5 \times 10^{-12} sec$. The propagation delays of the components of the system are given in Figure 3.2. Assume that the wires do not exhibit any delay, and all registers are clocked with the same clock signal.

[4]

[continued on the following page]

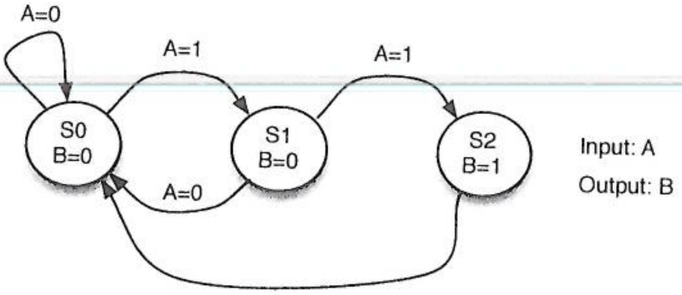


Figure 3.1 State diagram

symbol	value	description
T_p^{reg}	5ns	register propagation delay
T_{setup}^{reg}	2ns	register setup time
T_{hold}^{reg}	3ns	register hold time
T_p^{LUT}	10ns	4-LUT propagation time

Figure 3.2 Propagation delay of the components

4. The following function is required to be mapped into an FPGA:

$$f(x) = x^2 - 0.5x + 4$$

where x takes integer values in the range $[0, 255]$.

a) Derive an architecture for the function evaluation using only multipliers and adders that exhibits a throughput of one result per clock cycle. State the word-length (i.e. number of bits) of the busses in your design. The result should be evaluated at maximum possible accuracy. Assume that your input and output are registered and that the multipliers and adders cannot be pipelined.

[5]

b) Using Interval Arithmetic (IA), find the bounds for all the signals in your design.

[5]

c) Using the above estimated bounds through IA, derive the required word-length of the busses. Comment on the discrepancy, if any, between the derived word-length from part (a) and the estimated word-lengths derived using IA (part (b)).

[5]

d) Using the word-lengths for the busses from part (a), state the critical path of the design, and derive its maximum frequency. The propagation delays of the components are shown in Figure 4.1.

[5]

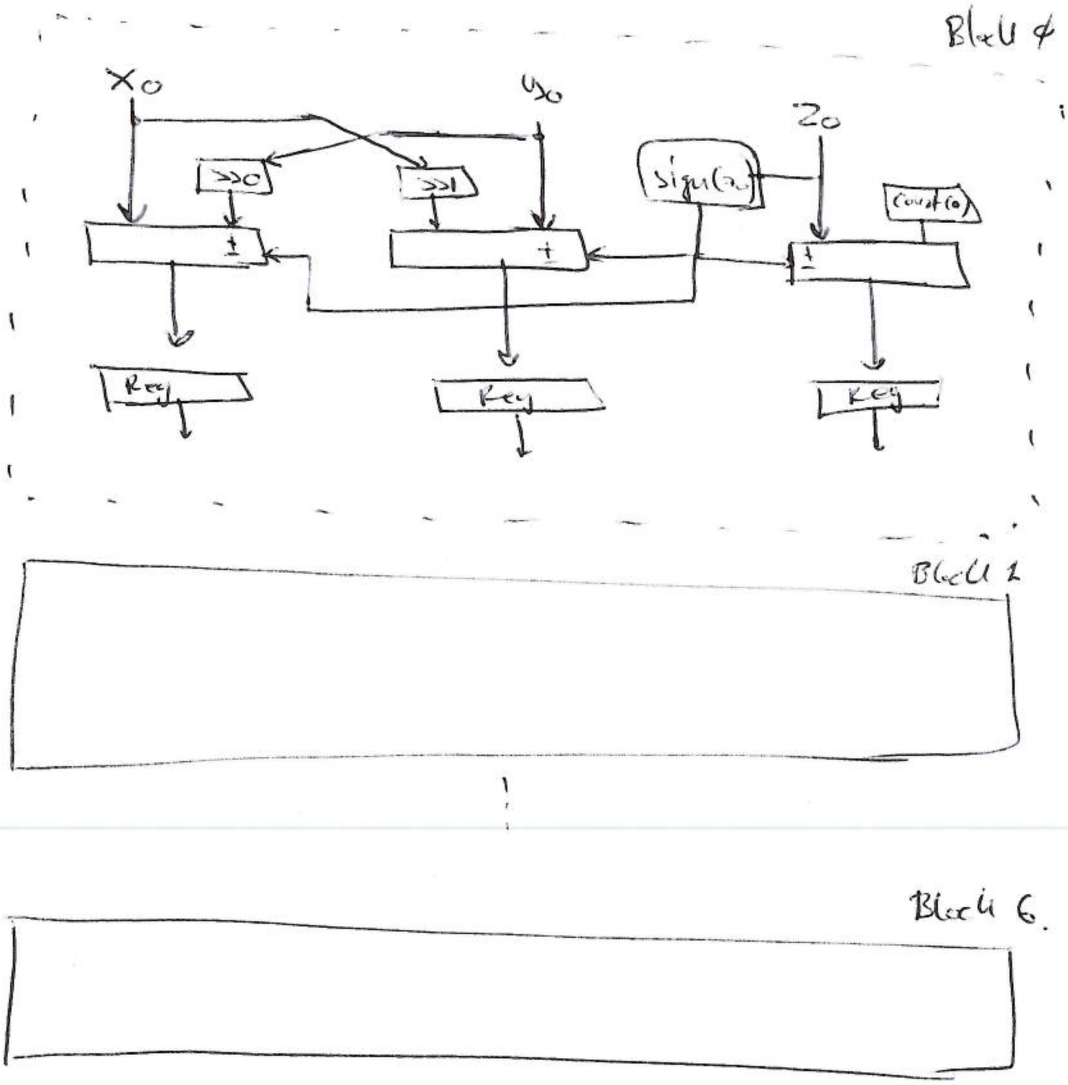
symbol	value	description
T_p^{reg}	5ns	register propagation delay
T_{setup}^{reg}	2ns	register setup time
T_{hold}^{reg}	3ns	register hold time
T_p^{mul}	$2.5m \times 1$ns	propagation time for a m x m bit parallel multiplier
T_p^{add}	$1.2m \times 1$ns	propagation time for a m-bit parallel adder

Figure 4.1 Propagation delay of the components

Master copy

1

a) [Bookwork + new application]



The only change is in the const(x) block and sign(zx) block, where x is the level. [6]

Throughput = 1 result per cycle [2]
latency = 7 cycles. [2]

[10]

1. b). [Bookwork]

2

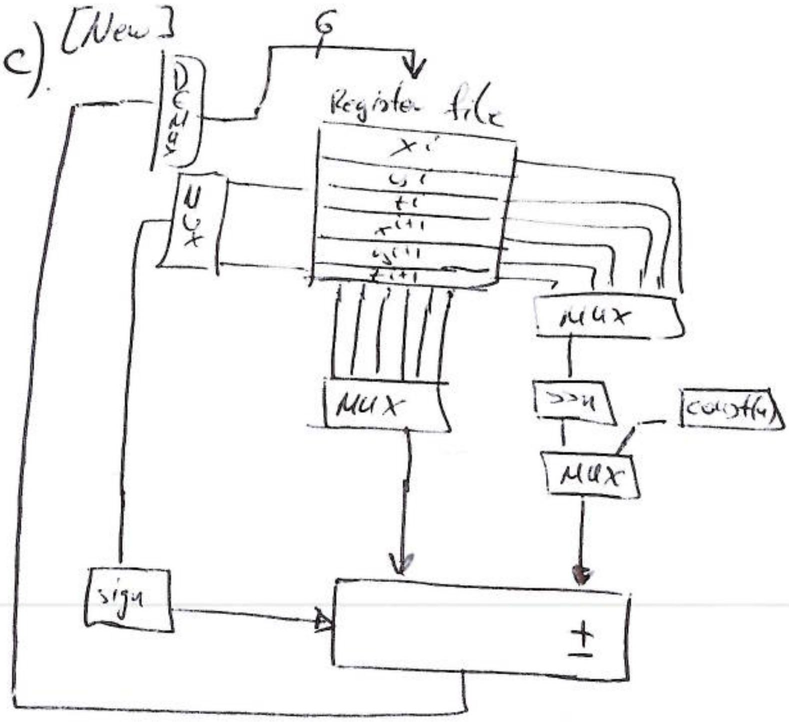
Initialise: $z=0$, $x=1$, $y=y$

Iterate with $d_i = \text{sign}(y_i)$

After n iterations $z_n = \arctan(y)$.

[5]

c). [New]

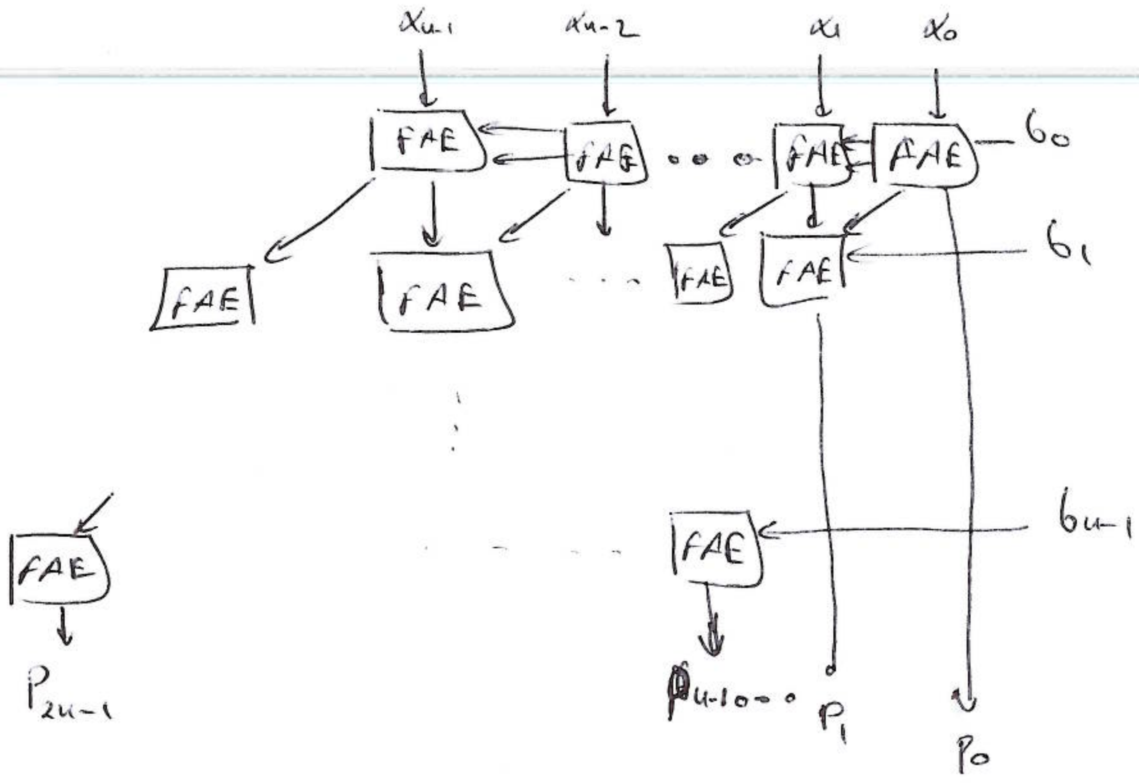


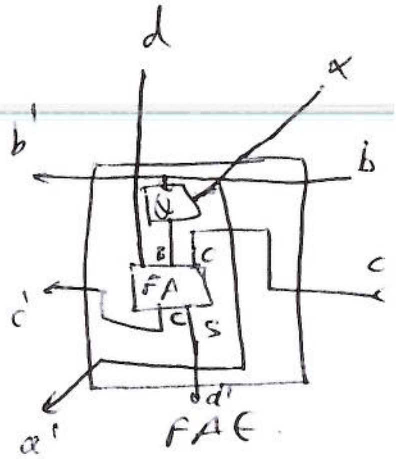
All muxes, demuxes, and shifts are controlled by an FSM

Throughput: ~~1 cy~~ 1 result per 7 cycles.

[5]

2. a). [Bookwork + new application]]





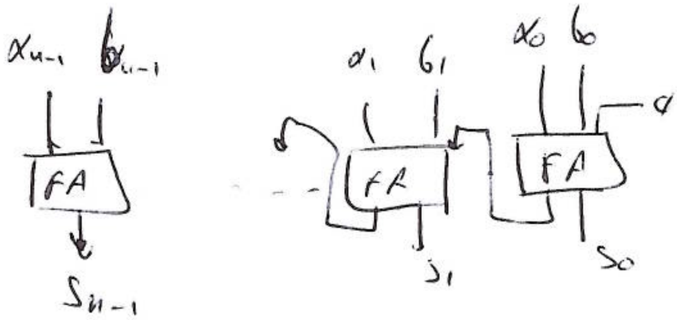
Each level i is the partial product of $x.b_i$.

The final result is the sum of all the partial products.

Resources = n^2

[7]

b)



Resources = n

[3]

2. c)

4

Parallel multiplier

Critical path: from the first FA to the left down corner.

Delay = $2n$

Adder

from the first FA to the last:

Delay $= n$

[5]

d)

The best solution is $14\cdot x = (16-2)\,x = 16\cdot x - 2\cdot x$.

$$\begin{array}{rll} & 16\cdot x & : \quad x_{n-1}\cdots x_1 x_0 0000 \\ - & 2\cdot x & - \; 0\,0\,x_{n-1}\cdots x_3 x_2 x_1 x_0 0 \\ \hline & & \underbrace{x_{n-1}\,x_{n-2}\,x_{n-3}\,r\,r\,r\dots 0}_{n+4 \text{ bits}} \end{array}$$

I need a single n bit sub unit.

Resource : n

Delay : n

Faster than the general multiplier (n vs. $2\cdot n$) and resources scale linearly with number of bits (n vs. n^2).

[5]

3). a) [New application]

5

3 states → 2 ff. ← [1]

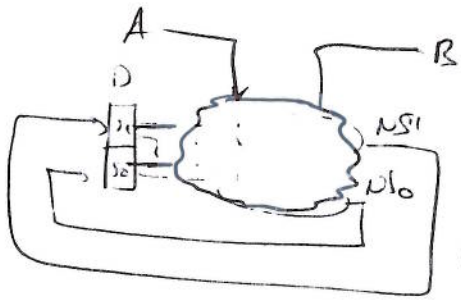
A	S_1	S_0	NS_1	NS_0	B
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	1
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	0	0	1

[2]

$$NS_1 = A \bar{S_1} S_0$$

$$NS_0 = A \bar{S_2} \bar{S_0}$$

$$B = \bar{A} S_1 \bar{S_0} + A S_1 \bar{S_0}$$ [2]



[2]

[7]

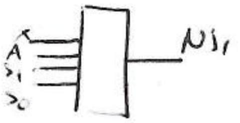
b) NS_1 : 1 LUT (3 input variable)

NS_0 : 1 LUT (3 " ")

B : 1 LUT (3 " ")

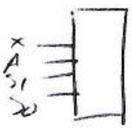
address

NS_1



	Not used.
111	X
110	0
101	1
100	0
011	X
010	0
001	0
000	0

NS_0



	Not used
111	X
110	0
101	0
100	1
011	x
010	0
001	0
000	0

B

	Not used
111	x
110	1
101	0
100	0
011	x
0Φ0	1
001	0
000	0

[6] :)

5

3)c) All the paths exhibit the same delay.

<u>Setup</u>:

$$t_p^{reg} + t_p^{LUT} + t_s^{reg} < T \Rightarrow 5+10+2 < T \Rightarrow \boxed{T > 17\,\text{us}}$$

<u>Hold</u> $\cancel{T} + t_p^{reg} + t_p^{LUT} > \cancel{T} + t_h \Rightarrow 5 + 10 > 3$ <u>always</u>

$$f_{max} = \frac{1}{17\,\text{usec.}}$$

[3]

d).

$$MTBF(t_r) = \frac{e^{t_r/c}}{T_0 \cdot f \cdot \alpha} \quad (1)$$

$$t_r = T - t_s^{reg} - t_p^{LUT} = 17 - 2 - 10 = 5\,\text{usec.}$$

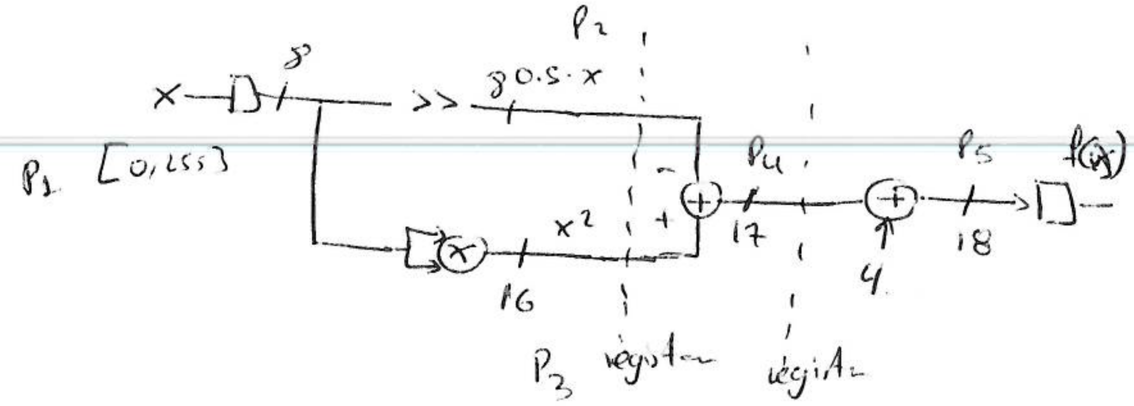
$$(1) \Rightarrow \alpha = \frac{e^{t_r/c}}{T_0 \cdot f \cdot MTBF} = \frac{e^{5 \cdot 10^{-4} / 1 \cdot 10^{-4}}}{9.5 \cdot 10^{-12} \cdot \frac{1}{17} \cdot 10^{+4} \cdot 3600} \pm$$

$\alpha =$ 77.77 trans/sec.

[4]

4. a) [new application]

7



[5]

b) P_i : denotes the points.

P_1 : [0, 255]

P_2 : [0, 127.5]

P_3 : [0, 65025]

P_4 : [0 − 127.5 , 65025 − 0] = [−127.5 , 65025]

P_5 : [−123.5 , 65029]

[5]

c) $P_1 \rightarrow$ 8 bits

$P_2 \rightarrow$ 8 bits

$P_3 \rightarrow \lceil \log_2 (65025+1) \rceil = 16$

$P_4 \rightarrow$ [illegible] 17

with ~~17~~ : −65015 → 65015 , +1 bit for 0.5 → 18 b

$P_5 \rightarrow$ 18 bits . Difference in P_4 due to inability of IA to i

nature correlations. [5

4. d)

8/8

The critical path is through the generic multiplier.

<u>Setup</u>

$$\left.\begin{aligned} t_p^{reg} + t_p^{mul} + t_s^{reg} &< T \\ t_p^{mul} &= 2.5 \cdot 8 \end{aligned}\right\} \Rightarrow$$

$$5 + 20 + 2 < T \Rightarrow T > 27 \text{ nsec.}$$

<u>Hold</u>

$$\cancel{T} + t_p^{reg} + t_p^{mul} > \cancel{T} + t_h^{reg} \Rightarrow$$

$$5 + 20 > 3 \quad \checkmark$$

$$f_{max} = \frac{1}{27 \text{ nsec}} \qquad [5].$$