

---

UNIVERSITI SAINS MALAYSIA

First Semester Examination  
Academic Session 2007/2008

October/November 2007

**EEE 348 – INTRODUCTION TO INTEGRATED CIRCUIT DESIGN**  
***[Pengantar Rekabentuk Litar Bersepadu]***

Duration: 3 hours  
*[Masa: 3 jam]*

---

Please check that this examination paper consists of EIGHT pages of printed material before you begin the examination.

*[Sila pastikan bahawa kertas peperiksaan ini mengandungi LAPAN muka surat bercetak sebelum anda memulakan peperiksaan ini].*

This paper contains SIX questions.

*[Kertas soalan ini mengandungi ENAM soalan].*

**Instructions:** Answer **FIVE** (5) questions. If a candidate answer more than five questions, only the first five answered will be examined and awarded marks.

**Arahan:** Jawab **LIMA** soalan. Jika calon menjawab lebih daripada lima soalan hanya lima soalan pertama mengikut susunan dalam skrip jawapan akan diberi markah].

Answer to any question must start on a new page.

*[Mulakan jawapan anda untuk setiap soalan pada muka surat yang baru].*

You may answer a question either in bahasa Malaysia or in English.

*[Anda dibenarkan menjawab soalan sama ada dalam bahasa Malaysia atau bahasa Inggeris].*

1. (a) Apakah Hukum Moore?  
*What is Moore's Law?* (3 marks)
- (b) Terangkan tiga bukti mengenai Hukum Moore.  
*Give three empirical evidence of Moore's Law.* (7 marks)
- (c) Terangkan asas aliran rekabentuk VLSI.  
*Explain basic VLSI design flow.* (4 marks)
- (d) Sekiranya anda merekabentuk litar bersepadu yang kompleks di mana ia memiliki fungsi analog dan digital, terangkan pilihan kaedah rekabentuk anda.  
*If you were to design a complex integrated circuit which consists of analog and digital function, explain your choice of design methodology.* (6 marks)
2. (a) Terangkan ciri-ciri penyongsang CMOS.  
*Explain the characteristics of standard CMOS inverter.*
- (i) Masa naik  
*Rise time* (3 marks)
- (ii) Masa turun  
*Fall time* (3 marks)
- (iii) Lengah perambatan  
*Propagation delay* (4 marks)
- (b) Anggaran lengah perambatan untuk penyongsang dengan menggunakan pengayun cincin adalah salah satu kaedah yang biasa.  
*Estimation of propagation delay of inverter from ring oscillator circuit is one of the common methods.*

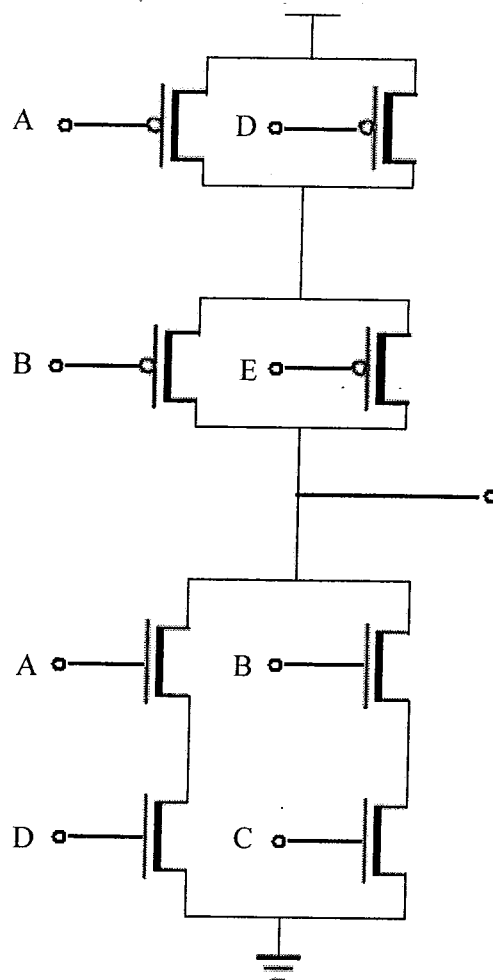
(i) Lukis pengayun cincin.  
*Draw ring oscillator.* (4 marks)

(ii) Terangkan bagaimana lengah perambatan penyongsang dapat ditentukan dengan menggunakan pengayun cincin.

*Explain how we can determine the propagation delay of an inverter from the ring oscillator.*

(6 marks)

3.



Gambarajah 1 Litar CMOS bergabungan  
*Figure 1 Combinational CMOS circuit*

...4/-

- (a) Dengan menggunakan laluan Euler sepunya, lukiskan gambarajah liti litar di dalam Gambarajah 1.

*By employing common Euler path, draw the stick diagram of the circuit in Figure 1.*

(11 marks)

- (b) Tentukan nisbah W/L bagi penyongsang yang menggunakan perintang sebagai beban seperti di dalam Gambarajah 2.  $R_L = 1\text{ k}\Omega$  dan  $V_{ol} = 0.6\text{ V}$  adalah kriteria litar tersebut. Transistor NMOS mempunyai parameter seperti di bawah.

*Determine the required aspect ratio, W/L of the resistive-load inverter in Figure 2.  $R_L = 1\text{ k}\Omega$ ,  $V_{ol} = 0.6\text{ V}$ . NMOS transistor has the following parameters.*

(9 marks)

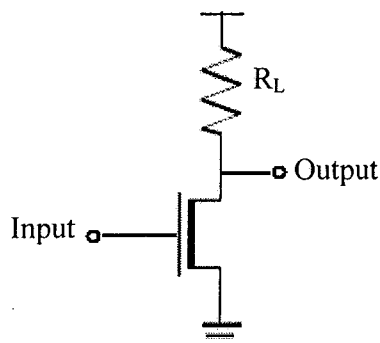
$$V_{DD} = 5.0\text{ V.}$$

$$V_{TO} = 0.9\text{ V.}$$

$$\gamma = 0.2\text{ V}^{1/2}$$

$$\lambda = 0.0\text{ V}^{-1}$$

$$k' = 22\text{ }\mu\text{A/V}^2$$



Gambarajah 2 Penyongsang dengan perintang sebagai beban  
*Figure 2 Resistive load inverter*

...5/-

4. (a) Huraikan pendekatan rekabentuk berikut:-

*Elaborate the following design approaches:-*

- (i) Kustom Penuh  
*Full-Custom*
- (ii) Kustom Separa  
*Semi-Custom*
- (iii) FPGA

(8 marks)

- (b) Bezakan, dengan menunjukkan contoh kod, pendekatan pengkodan Verilog berikut dari segi keutamaannya.

*Differentiate, by showing a sample code, the following coding approach in terms of its significance.*

- (i) Pengkodan Kelakuan  
*Behavioural Coding*
- (ii) Pengkodan Struktur  
*Structural Coding*

(8 marks)

- (c) Kod Verilog berikut adalah bagi menambah dua integer x dan y bertanda. Hasilnya adalah z yang sengaja dijadikan satu bit lebih lebar. Apakah hasil desimal bertanda bagi z selepas kod ini dilaksanakan?

*The following Verilog code is intended to add two signed integers x and y. The result is z, which is intentionally made one bit wider. What is the numerical signed decimal result in z after running the code?*

Kod Verilog:  
*Verilog code:*

```
reg [3:0] x, y; reg [4:0] z;  
assign x=-4'd3;  
assign y=4'd5;  
assign z=x+y;
```

(4 marks)

...6/-

5. (a) Bezakan di antara kod Verilog boleh-sintesis dengan tak boleh-sintesis dari segi kegunaannya.

*Differentiate between synthesis able and a non-synthesis able Verilog code, in terms of its use.*

(4 marks)

- (b) Rantaian 'for' bagi kod di bawah kelihatan tiada masalah, sebaliknya adalah rantaian infinit. Mengapa?

*The 'for' loop in the code below looks harmless but is actually an infinite loop. Why?*

```
module iloop(z,a);  
input [31:0] a;  
output z;  
reg [4:0] i;  
reg s, z;  
initial begin  
s = 0;  
for(i=0; i<32; i=i+1) s = s | a[i];  
z = !s;  
end  
endmodule
```

(4 marks)

- (c) Isikan nilai-nilai bagi a, b, c, d dan e dalam jadual seperti di bawah ini. Bagi setiap baris hanya isikan nilai keadaan mantap. Contohnya sekiranya nilai-nilai dalam jadual tidak berubah untuk masa 3 dan seterusnya, hanya isi bagi baris 0 hingga 2. Sekiranya satu nilai diumpuk beberapa kali bagi masa simulasi yang sama, hanya tunjukkan nilai akhir bagi masa simulasi tersebut.

*Fill in values for a, b, c, d, and e in the following table. Only fill in rows of the table until all values reach their steady state values. For example, if the values in the table do not change for time 3 and beyond, only fill in the rows for time 0 through time 2. If a value is assigned multiple times within the same simulation time, only show its final value for that simulation time.*

```
module timing;  
reg [4:0] a, b, c, d, e;  
reg clk;  
initial begin c = 3; d = 2, e = 4; #2 clk = 1; end  
always@(a, b) begin  
#1 c = #1 a + 2; // statement 1  
d = #1 c + b; // statement 2  
end  
always @(posedge clk) begin  
#2 e <= c + d; // statement 3  
a <= #1 e - d; // statement 4  
b <= #1 e + d; // statement 5  
end  
end module
```

Time	a	b	c	d	e
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					

(12 marks)

6. Aliran rekabentuk ASIC bermula dari spesifikasi dan berakhir dengan 'tapeout'. Di antaranya terdapat berbagai proses termasuk: Pengkodan RTL, Simulasi, Testbench', Sintesis, Analisis Pemasaan Pra-Bentangan dan Post-Bentangan, APR, Anotasi Belakang dan Penentusahan Logik. Dengan menggunakan rajah yang sesuai, huraikan kesemua proses yang diperlukan bagi menghasilkan rekabentuk dari spesifikasi ke 'tapeout'.

*An ASIC design flow starts from the specification and end with the tapeout. In between, some of the processes include: RTL coding, Simulation, Testbench, Synthesis, Pre-Layout and Post-Layout Timing Analysis, APR, Back Annotation and Logic Verification. Using relevant diagram, elaborate on all the required processes to get the design from specification up to tapeout.*

(20 marks)

oooOOooo