Second Semester Examination
2018/2019 Academic Session

June 2019

EEE348 – Introduction to Integrated Circuit Design
(Pengantar Rekabentuk Litar Bersepadu)

Duration : 3 hours
(Masa : 3 jam)

Please ensure that this examination paper consists of TWELVE (12) pages and ONE (1) page of printed appendix material before you begin the examination.

[Sila pastikan bahawa kertas peperiksaan ini mengandungi DUA BELAS (12) muka surat dan SATU (1) muka surat lampiran yang bercetak sebelum anda memulakan peperiksaan ini.]

Instruction: This question paper consists of **FIVE (5)** questions. Answer **ALL** questions. All questions carry the same marks.

***[Arahan:** Kertas soalan ini mengandungi **LIMA (5)** soalan. Jawab **SEMUA** soalan. Semua soalan membawa jumlah Markah yang sama.]*

In the event of any discrepancies, the English version shall be used.

[Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi Bahasa Inggeris hendaklah digunapakai.]

...2/-

1. (a) Figure 1 below shows the layout mask sets of a CMOS logic circuit.

Rajah 1 di bawah menggambarkan set topeng untuk susun atur sebuah logik CMOS.

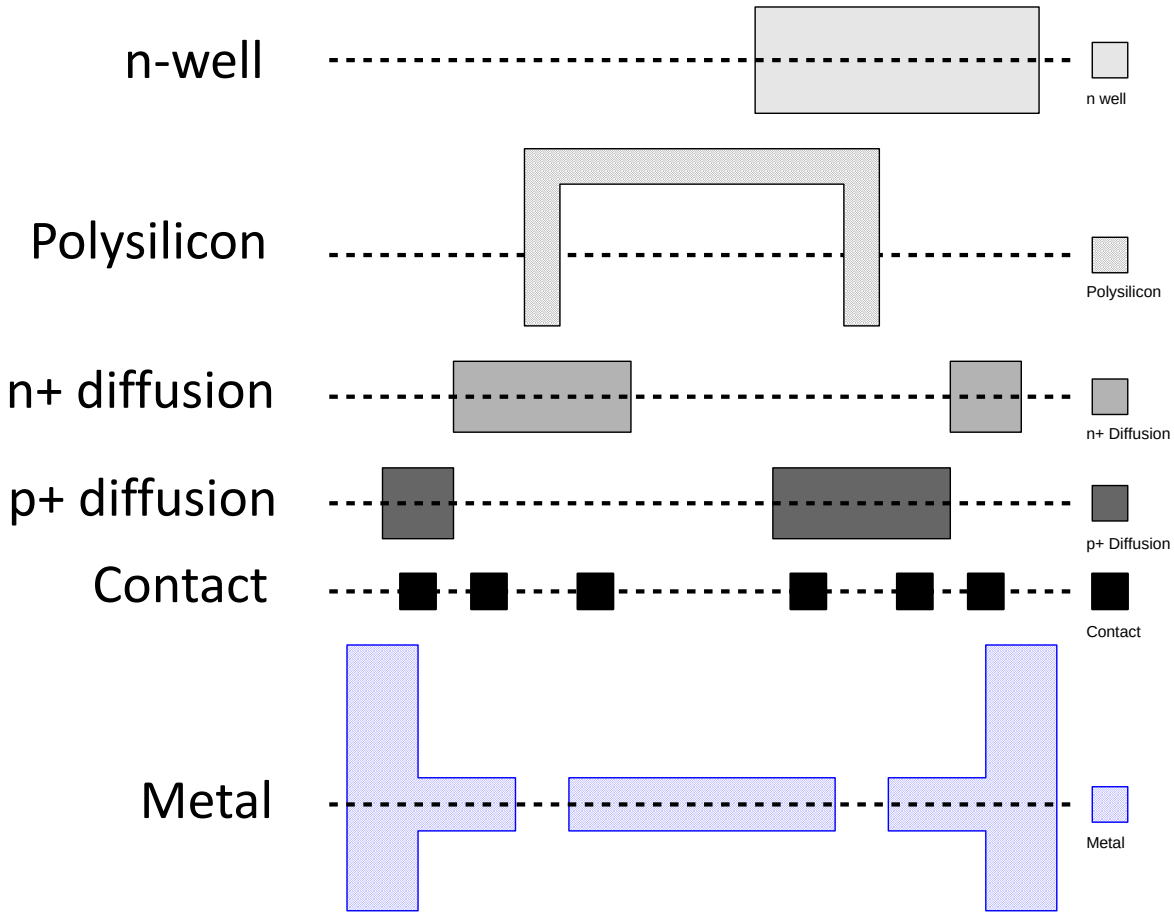


Figure 1
Rajah 1

(i) Draw and label the equivalent layout for the mask sets above.
Lukis dan labelkan gambarajah susun atur untuk set topeng di atas.

(20 marks/*markah*)

(ii) Draw the equivalent schematic.
Lukis gambarajah litar.

(10 marks/*markah*)

…3/-

(iii) Draw and label the corresponding cross section view.

Lukis dan labelkan keratan rentas sepadan.

(20 marks/*markah*)

(b) The truth table below represents the operation of a CMOS logic.

Jadual kebenaran di bawah mewakili operasi sebuah logik CMOS.

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

(i) Draw the schematic
Lukis gambarajah litar

(20 marks/*markah*)

(ii) Proof the logic operation for each input condition tabulated in the truth table above is correct using the **switching diagram**.

Buktikan operasi logik untuk setiap keadaan masukan yang dinyatakan dalam jadual di atas dengan menggunakan ***gambarajah suis***.

(20 marks/*markah*)

(c) Illustrate the general design flow for full custom methodology.

Gambarkan aliran rekabentuk umum untuk kaedah full custom.

(10 marks/*markah*)

2. (a) The Boolean Function of a complex logic gate is given as below:

Fungsi Boolean untuk sebuah gate logik adalah seperti berikut:

$$F = \overline{(AB + C)(D + E)}$$

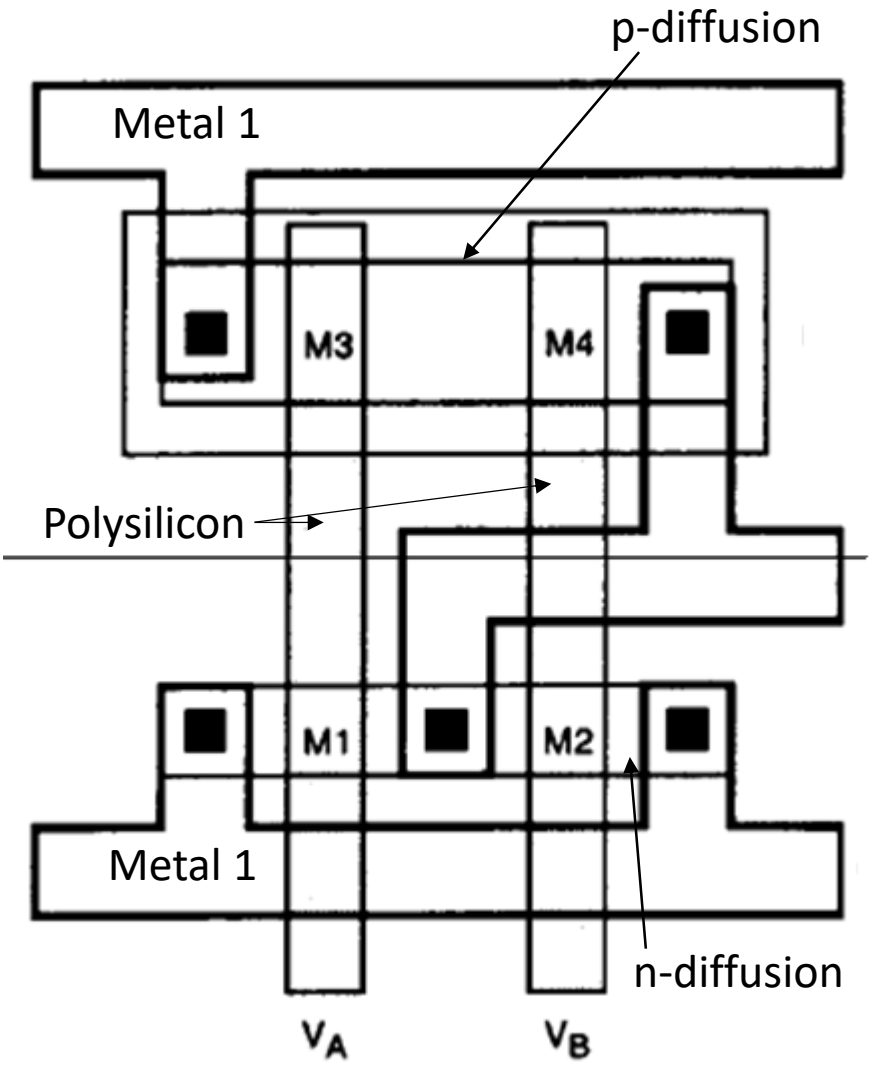
Construct the schematic of the logic circuit using the Pull Up and Pull Down Networks.

Bina litar skematik dengan menggunakan rangkaian Pull Up dan Pull Down.

(30 marks/*markah*)

…4/

(b) Figure below illustrates the layout of a CMOS logic gate.
Gambarajah di bawah menunjukkan susun atur sebuah logik CMOS.



(i) Draw the Stick Diagram
Lukiskan gambarajah lidi.

(30 marks/*markah*)

(ii) Draw the schematic
Lukiskan litar skematik.

(20 marks/*markah*)

(c) Draw the schematic diagram of a Dynamic RAM (DRAM)
Lukiskan gambarajah skematik Dynamic RAM (DRAM)

(20 marks/*markah*)

3. (a) Design a digital circuit based on following specification using Verilog HDL. You have to use the concept of connection of instances for the top level module, where the top level module only consists of the instances of low level module. As for the low level module, you may use any type of description. You need to develop also the test bench to verify the designed circuit. No need to test all combinations.

The circuit has two inputs ***t*** and ***in***, where ***t*** is an 8-bit binary number and ***in*** is an integer in the range of 0 to 15. An output ***out*** is based on the total number of "0" in the input ***t*** as follows:-

If total number of "0" is between 0 and 2, ***out*** = 2*****in***
If total number of "0" is between 3 and 4, ***out*** = 4***in***
If total number of "0" is between 5 and 6, ***out*** = 6***in***
If total number of "0" is between 7 and 8, ***out*** = 8***in***

Rekabentuk sebuah litar digital berdasarkan spesifikasi berikut dengan menggunakan Verilog HDL. Anda dikehendaki menggunakan konsep "connection of instances" untuk modul peringkat atasan, yang mana modul peringkat atasan hanya mengandungi beberapa "instances" bagi modul peringkat bawahan. Untuk modul peringkat bawahan, anda boleh menggunakan mana-mana jenis deskripsi. Anda juga dikehendaki membangunkan "test bench" untuk mengesahkan litar yang telah direka. Tidak perlu untuk membuat pengesahan untuk kesemua kemungkinan.

Litar tersebut mempunyai dua masukan iaitu ***t*** dan ***in****, di mana* ***t*** *adalah nombor binari 8-bit dan* ***in*** *adalah nombor integer di antara 0 hingga 15. Keluaran* ***out*** *adalah berdasarkan jumlah nombor "0" di dalam masukan* ***t*** *seperti berikut:-*

Sekiranya jumlah nombor "0" adalah di antara 0 dan 2, ***out*** *= 2*****in***
Sekiranya jumlah nombor "0" adalah di antara 3 dan 4, ***out*** *= 4*****in***
Sekiranya jumlah nombor "0" adalah di antara 5 dan 6, ***out*** *= 6*****in***
Sekiranya jumlah nombor "0" adalah di antara 7 dan 8, ***out*** *= 8*****in***

…6/-

The example are as follows :-

Contoh adalah seperti berikut:-

If ***t*** = 00001111 and ***in*** = 5, ***out*** = 4*5
If ***t*** = 11001000 and ***in*** = 14, ***ou***t = 6*14

*Sekiranya **t** = 00001111 dan **in** = 5, **out** = 4*5*
*Sekiranya **t** = 11001000 dan **in** = 14, **out** = 6*14*

(80 marks/*markah*)

(b) Design a digital circuit based on following specification using Verilog HDL.

Rekabentuk sebuah litar digital berdasarkan spesifikasi berikut dengan menggunakan Verilog HDL.

(i) There are three inputs (***in1***, ***in2*** and ***in3***). Each of the inputs is an integer number in the range of 0 to 15.

*Terdapat tiga masukan (**in1**, **in2** dan **in3)**. Setiap masukan adalah nombor integer di antara 0 hingga 15.*

(ii) An output ***out*** is based on following conditions:-

*Keluaran **out** adalah berdasarkan situasi berikut:-*

If ***max(in1, in2, in3)*** is an even number, ***out = 1***

*Sekiranya **max(in1, in2, in3)** adalah nombor genap, **out = 1***

If ***max(in1, in2, in3)*** is an odd number, ***out = 0***

*Sekiranya **max(in1, in2, in3)** adalah nombor ganjil, **out = 0***

(iii) You have to use the concept of continuous assignment and no need to write the test bench.

Anda dikehendaki menggunakan konsep "continuous assignment" dan tidak perlu menulis "test bench".

(20 marks/*markah*)

…7/-

4. (a) Consider following Verilog modules. Draw the waveform for the input signal CLK and the output signal OUT.

Diberi modul-modul Verilog seperti berikut. *Lukis gambarajah gelombang untuk isyarat masukan CLK dan juga isyarat keluaran OUT.*

```
module q4a(out, clk, rst, in0, in1);
 output out;
 input clk, rst;
 input [7:0] in0, in1;
 reg [7:0] t0, t1;
 reg [2:0] a, b;

   always @(posedge clk)
      if (rst == 0)
        begin
           t0 <= in0;
           t1 <= in1;
        end

   always @(t0, t1)
     begin
      casex(t0)
         8'b0xxxxxxx: a = 7;
         8'bx0xxxxxx: a = 6;
         8'bxx0xxxxx: a = 5;
         8'bxxx0xxxx: a = 4;
         8'bxxxx0xxx: a = 3;
         8'bxxxxx0xx: a = 2;
         8'bxxxxxx0x: a = 1;
         8'bxxxxxxx0: a = 0;
         default: a = 3'bxxx;
      endcase
      casex(t1)
```

…8/-

```
            8'b0xxxxxxx: b = 7;
            8'bx0xxxxxx: b = 6;
            8'bxx0xxxxx: b = 5;
            8'bxxx0xxxx: b = 4;
            8'bxxxx0xxx: b = 3;
            8'bxxxxx0xx: b = 2;
            8'bxxxxxx0x: b = 1;
            8'bxxxxxxx0: b = 0;
            default: b = 3'bxxx;
        endcase
       end
     assign out = a>b;
endmodule

module tb_q4a;
    wire OUT;
    reg CLK, RST;
    reg [7:0] IN0, IN1;

    q4a dut(OUT, CLK, RST, IN0, IN1);

    always
      #1 CLK = ~CLK;
    initial begin
       CLK = 0; RST = 1;
       IN0 = 8'b11100001; IN1 = 8'b00000000;
       #2 IN0 = 8'b00000111; IN1 = 8'b11111111;
       #2 RST = 0; IN0 = 8'b01000111; IN1 = 8'b11110111;
       #2 IN0 = 8'b11110111; IN1 = 8'b00111111;
       #2 IN0 = 8'b11000111; IN1 = 8'b11100111;
       #2 IN0 = 8'b11111011; IN1 = 8'b10110111;
       #2 IN0 = 8'b11000111; IN1 = 8'b01111110;
       #2 IN0 = 8'b11100101; IN1 = 8'b11111110;
       #2  $stop;
    end
endmodule
```

(50 marks/*markah*)

(b) Consider the following poorly-written SystemVerilog code and answer the subsequent questions.

Pertimbangkan kod SystemVerilog yang ditulis dengan lemah berikut dan jawab soalan-soalan berikutnya.

```
module q4b (i_clk, i_rst_b, R1, R2, R3, Y1, Y2, Y3, G1, G2, G3, G4, up, cnt);

inputi_clk, i_rst_b, up, cnt;
output logic R1, R2, R3, Y1, Y2, Y3, G1, G2, G3, G4;

parameter P = 3'b000, Q = 3'b001, R = 3'b010, S = 3'b011;
parameter T = 3'b100, U = 3'b101, V = 3'b110, W = 3'b111;

logic [2:0] FSM_ps, FSM_ns;
logic a, b, c, d, e, f, g, h, i;

assign a = up & ~cnt;
assign b = up;
assign c = ~up;
assign d = up & cnt;
assign e = up;
assign f = up;
assign g = up;
assign h = up;
assign i = ~up;

assign R1 = (FSM_ps==P) | (FSM_ps==Q) | (FSM_ps==R);
assign R2 = (FSM_ps==P) | (FSM_ps==Q) | (FSM_ps==R) | (FSM_ps==U) |
            (FSM_ps==V) | (FSM_ps==W);
assign R3 = FSM_ps==P) | (FSM_ps==S) | (FSM_ps==T) | (FSM_ps==U) |
            (FSM_ps==V) | (FSM_ps==W);
assign Y1 = (FSM_ps==W);
assign Y2 = (FSM_ps==T);
assign Y3 = (FSM_ps==R);
assign G1 = (FSM_ps==S) | (FSM_ps==T) | (FSM_ps==U) | (FSM_ps==V);
assign G2 = (FSM_ps==S);
assign G3 = (FSM_ps==Q);
assign G4 = (FSM_ps==V);
```

```
    always @(FSM_ps or a or b or c or d or e or f or g or h or i)
    begin: FSM_fsm
case (FSM_ps)
          P:begin if (a) begin FSM_ns = Q; end
                else if (d) begin FSM_ns = S; end
          else begin FSM_ns = P; end end

 Q:begin if (b) begin FSM_ns = R; end
                else begin FSM_ns = Q; end end

 R:begin if (c) begin FSM_ns = P; end
                else begin FSM_ns = R; end end

 S:begin if (e) begin FSM_ns = T; end
   else begin FSM_ns = S; end end

 T:begin if (f) begin FSM_ns = U; end
                  else begin FSM_ns = T; end end

 U:begin if (g) begin FSM_ns = V; end
   else begin FSM_ns = U; end end

 V:begin if (h) begin FSM_ns = W; end
                else begin FSM_ns = V; end end

 W:begin if (i) begin FSM_ns = P; end
                else begin FSM_ns = W; end end

          default: begin FSM_ns = 3'bxxx; end
endcase
    end

    always_ff @ (posedge i_clk or negedge i_rst_b)
    begin: FSM_state_flops
if (i_rst_b == 1'b0) begin FSM_ps <= #100 P; end
          else begin FSM_ps <= #100 FSM_ns; end
    end

    endmodule
```

(i) Provide good coding practices to ease code readability, understandability, and troubleshooting.

Berikan amalan pengekodan baik bagi memudahkan pembacaan, pemahaman, dan pemulihan kod.

(10 marks/*markah*)

(ii) Based on the given SystemVerilog code, draw the Finite State Machine (FSM).

Berdasarkan kod SystemVerilog yang diberikan, lukis Mesin Keadaan Terhingga (FSM).

(40 marks/*markah*)

5. A digital circuit is to be designed with three 16-bit registers RA, RB, and RC that perform the following operations:

Sebuah litar digital perlu direka bentuk dengan tiga pendaftar 16-bit RA, RB, dan RC yang melaksanakan operasi-operasi berikut:

(i) Transfer two 16-bit signed numbers A and B (in 2's-complement format) to registers RA and RB, respectively;

Pindahkan dua nombor bertanda 16-bit A dan B (dalam format 2-pelengkap) masing-masing ke pendaftar RA dan RB;

(ii) If the number in RA is negative, divide this number by 2 and transfer the result to RC;

Jika nombor dalam RA ialah negative, bahagi nombor ini dengan 2 dan pindahkan hasilnya ke RC;

(iii) If the number in RA is positive but nonzero, multiply the number in RB by 2 and transfer the result to RC;

Jika nombor dalam RA ialah positif tetapi bukan sifar, darab nombor dalam RB dengan 2 dan pindahkan hasilnya ke RC;

(iv) f the number in RA is zero, clear register RC to 0;
Jika nombor dalam RA ialah sifar, kosongkan pendaftar RC ke 0;

(v) A multiplier-free and divider-free design, i.e. the multiplication and division operations are performed without using any multiplier or divider.

Sebuah reka bentuk bebas pendarab dan bebas pembahagi, iaitu operasi-operasi darab dan bahagi dilaksanakan tanpa menggunakan sebarang pendarab atau pembahagi.

(a) Plan the Register Transfer Level (RTL) notation of the circuit.

Rancangkan notasi Peringkat Pemindahan Pendaftar (RTL) bagi litar berkenaan.

(25 marks/*markah*)

(b) Design the system block diagram of the circuit, consisting of Control Unit (CU) and Datapath Unit (DU).

Reka bentuk gambar rajah blok sistem litar berkenaan, terdiri daripada Unit Kawalan (CU) dan Unit Laluan Data (DU).

(35 marks/*markah*)

(c) Design the Algorithmic State Machine and Datapath (ASMD) of the circuit.

Reka bentuk Mesin Keadaan Algoritma dan Laluan Data (ASMD) bagi litar berkenaan.

(40 marks/*markah*)

-oooOooo-

APPENDIX

LAMPIRAN

Course Outcomes (CO) – Programme Outcomes (PO) Mapping

Pemetaan Hasil Pembelajaran Kursus – Hasil Program

Questions ***Soalan***	**CO**	**PO**
1	2	1
2	2	1
3	3	1
4	3	1
5	3	1
6		