



## มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

การสอบกลางภาค, 2/2560

เลขที่นั่งสอบ

วิชา ENE 231 Digital Circuit and Logic Design

สำหรับ นศ. ปี 2 ภาควิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

สอบ อังคารที่ 6 มี.ค. พ.ศ. 2561

เวลา 13:00น - 16:00น

### คำสั่ง

ky 13/3/61

1. ข้อสอบมีทั้งหมด 4 ข้อ 15 หน้า (รวมใบปะหน้า) คะแนนรวมทั้งหมด 95 คะแนน
2. ไม่อนุญาตให้นำหนังสือประกอบการเรียน และเอกสารที่เกี่ยวข้องกับวิชาเข้าห้องสอบ
3. แสดงวิธีทำลงในข้อสอบเท่านั้น (ถ้าพื้นที่ไม่พอให้ใช้ด้านหลังของข้อสอบได้)
4. ไม่อนุญาตให้ใช้เครื่องคิดเลขทุกชนิด
5. ไม่อนุญาตให้นำพจนานุกรมเข้าห้องสอบ
6. ห้ามนำข้อสอบและกระดาษคำตอบออกนอกห้องสอบ

### คำเตือนและคำแนะนำ

- เมื่อนักศึกษาทำข้อสอบเสร็จก่อนเวลา ต้องยกมือบอกกรรมการคุมสอบ เพื่อขออนุญาตออกนอกห้องสอบ
- นักศึกษาซึ่งทุจริตในการสอบ อาจถูกพิจารณาโทษสูงสุดให้พ้นสภาพการเป็นนักศึกษา
- นักศึกษาควรดูข้อสอบทั้งหมดก่อนเริ่มลงมือทำและควรอ่านคำถามให้รอบคอบก่อนเริ่มทำการคำนวณ เพื่อไม่ให้เสียเวลากับการคำนวณที่ไม่มีประโยชน์

ทำให้สบาย!!!!ไม่ทุจริต!!!!ขอให้โชคดีในการสอบ!!!

ผศ. ดร. พินิจ กำหอม (โทร. 9075, 9070) ผู้ออกข้อสอบ

ข้อสอบนี้ได้ผ่านการประเมินจากคณะกรรมการประจำภาคแล้ว

(รองศาสตราจารย์. ดร. ราชวดี ศิลาพันธ์)

หัวหน้าภาควิชาวิศวกรรมอิเล็กทรอนิกส์และโทรคมนาคม

ชื่อ-สกุล:							รหัสนักศึกษา:				
Prob. No.	1	2	3	4	5	6	7	8	9	10	Total
Full Score	20	30	35	10							95
Recieved Score											

1. (20 คะแนน) Information Representation, Digital Signals, and Radix Numbers
  - 1.1.(4 คะแนน) พิจารณาสารสนเทศที่เป็นคำตอบของคำถามต่อไปนี้ เพื่อระบุเซตของตัวเลขที่เหมาะสมในการแทนสถานะของสารสนเทศ ให้กำหนดช่วงและความละเอียดของตัวเลขเองตามความเหมาะสม
    - a. อุณหภูมิบริเวณผิวของ IC ตัวนี้ ณ เวลาที่วัด เป็นเท่าไรในหน่วยเซลเซียส
    - b. ค่ากระแสไฟฟ้า AC 220V ที่ไหลผ่านสายไฟเส้นนี้ ณ เวลาที่วัดเป็นกี่แอมป์
    - c. รถยนต์คันนี้กำลังอยู่ในสถานะอะไร? จอดอยู่, วิ่งอยู่, หรือ เข้าอู่ซ่อม
    - d. จากสัญญาณเสียงที่ได้ยินคน ๆ นี้สั่งงานให้ทำอะไร? เปิดไฟ ปิดไฟ เปิดแอร์ ปิดแอร์ หรือ ไม่ใช้การสั่งงาน
  - 1.2.(6 คะแนน) จากโดเมนของสถานะของสารสนเทศที่ระบุด้วยเซตของตัวเลขต่อไปนี้ ให้แทนสารสนเทศดังกล่าวด้วยสัญญาณดิจิทัล  $n$  บิต ด้วยวิธีการ offset เขียนสูตรความสัมพันธ์ระหว่างสัญญาณดิจิทัลกับสถานะของสารสนเทศ
    - a. สารสนเทศแทนด้วยเลขจำนวนจริงในช่วง  $-5$  ถึง  $5$  ด้วยความละเอียด  $0.001$

b. สารสนเทศแทนด้วยเลขจำนวนเต็มในช่วง -30 ถึง 100

1.3.(3 คะแนน) แสดงการบวกเลขฐานต่อไปนี้

a.  $22102.211_3 + 1212.012_3 = (\text{_____})_3$

b.  $22102.211_5 + 1212.012_5 = (\text{_____})_5$

c.  $21F2_{16} + 1A2B_{16} = (\text{_____})_{16}$

1.4.(7 คะแนน) แสดงการแปลงเลข 1125.325 เป็นเลขฐาน 2, ฐาน 8 และ ฐาน 16

$(1125.325)_{10} = (\text{_____})_2 = (\text{_____})_8 = (\text{_____})_{16}$

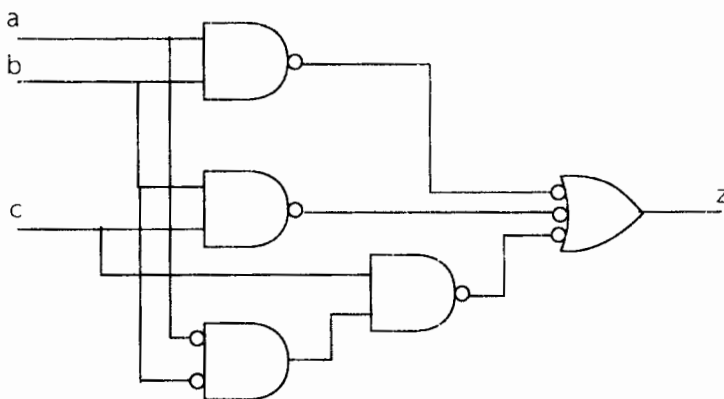
2. (30 คะแนน) Combination Systems and Boolean Algebra

2.1. (5 คะแนน) ให้ใช้ทฤษฎีของ Boolean Algebra แปลง logic expression ทางด้านซ้ายเป็น logic expression ทางด้านขวา

$$(a\bar{b} + a\bar{b}c + \bar{a})(a\bar{c} + ac + abc)(\overline{ab + c}) = a\bar{b}c$$

$$\begin{aligned} (a\bar{b} + a\bar{b}c + \bar{a})(a\bar{c} + ac + abc)(\overline{ab + c}) &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= \underline{\hspace{10em}} ( \hspace{1em} ) \\ &= a\bar{b}c \hspace{10em} ( \hspace{1em} ) \end{aligned}$$

2.2. (5 คะแนน) ให้วิเคราะห์หาตารางความจริงของ Combinational system ที่อธิบายด้วยวงจรลอจิกข้างล่างนี้ แสดงขั้นตอนการทำ



## 2.3. (20 คะแนน) จากตารางความจริงที่กำหนดให้

2.3.1. (1 คะแนน) เขียน canonical SOP logic expression ของ z

2.3.2. (1 คะแนน) เขียน canonical POS logic expression ของ z

2.3.3. (12 คะแนน) ใช้ Karnaugh Map Technique ในการหา minimal SOP โดยแสดงขั้นตอนการทำ

2.3.4. (12 คะแนน) ใช้ Quine-McCluskey (QM) technique ในการหา minimal POS โดยแสดงขั้นตอนการทำ

2.3.5. (4 คะแนน) เปรียบเทียบ complexity ของ canonical SOP, canonical POS, minimal SOP, และ minimal POS

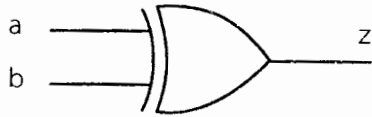


0000	0
0001	0
0010	1
0011	1
0100	0
0101	1
0110	0
0111	1
1000	0
1001	1
1010	1
1011	1
1100	0
1101	1
1110	0
1111	0

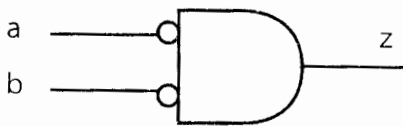
3. (35 คะแนน) Digital Building Blocks and VHDL

3.1.(15 คะแนน) จากสัญลักษณ์ของ Digital Building Blocks ที่ให้มาในแต่ละข้อ ให้บอกชื่อ และเขียนตารางแสดงการทำงาน (Function Table) ของมัน

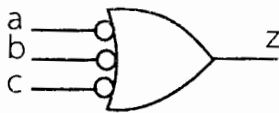
a. (1 คะแนน) \_\_\_\_\_



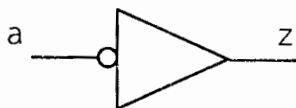
b. (1 คะแนน) \_\_\_\_\_



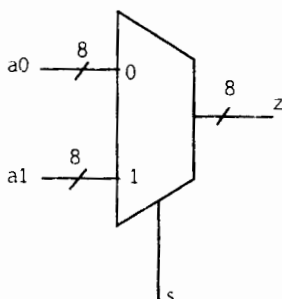
c. (2 คะแนน) \_\_\_\_\_



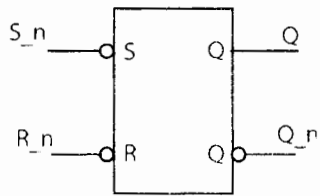
d. (1 คะแนน) \_\_\_\_\_



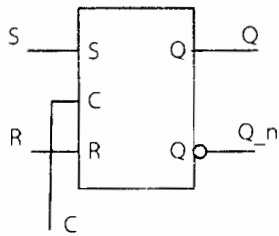
e. (2 คะแนน) \_\_\_\_\_



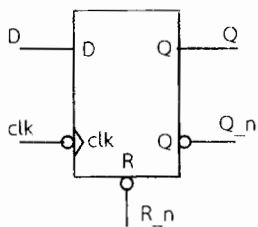
f. (2 คะแนน) \_\_\_\_\_



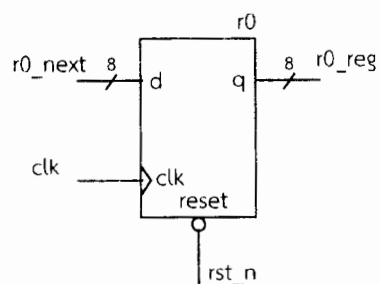
g. (2 คะแนน) \_\_\_\_\_



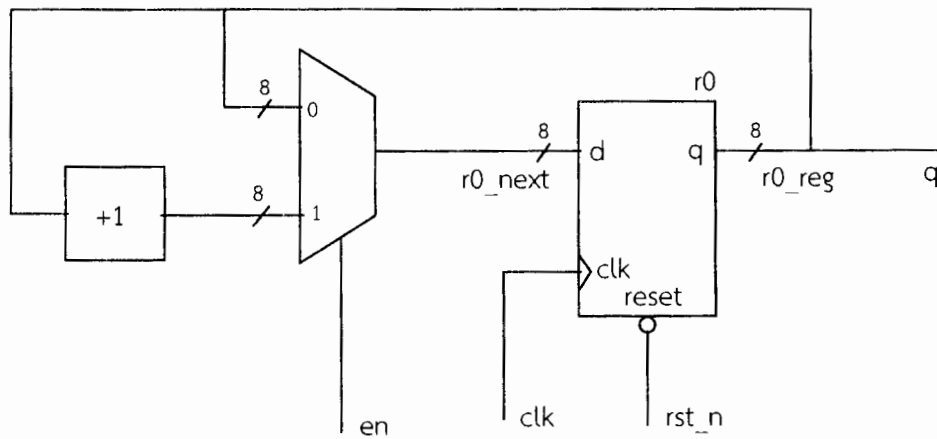
h. (2 คะแนน) \_\_\_\_\_



i. (2 คะแนน) \_\_\_\_\_



3.2.(10 คะแนน) จาก Block diagram อธิบายระบบดิจิทัลที่นำมาในแต่ละข้อ ให้เขียน VHDL program เพื่ออธิบายการทำงานของมัน





3.3.(10 คะแนน) จาก VHDL program อธิบายระบบดิจิทัลที่นำมา ให้เขียน Block diagram ของมัน

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity timer is
  port (
    clk, reset : in std_logic;
    t_p        : out std_logic;
    q          : out std_logic_vector(15 downto 0));
end entity timer;

architecture rtl of timer is
  constant INIT : unsigned(15 downto 0) := to_unsigned(40000-1,16);
  constant CNST : unsigned(15 downto 0) := to_unsigned(40000-1,16);
  signal r_reg, r_next : unsigned(15 downto 0);
  signal timer_p : std_logic;
begin -- architecture rtl

  process (clk, reset) is
  begin -- process
    if reset = '1' then
      r_reg <= INIT;
    elsif clk'event and clk = '1' then -- rising clock edge
      r_reg <= r_next;
    end if;
  end process;

  timer_p <= '1' when r_reg = 0 else
    '0';
  r_next <= CNST when timer_p = '1' else
    r_reg - 1;
  t_p <= timer_p;
  q <= std_logic_vector(r_reg);

end architecture rtl;
```

4. (10 คะแนน) [Digital System Design Methodology] ให้อธิบายกระบวนการออกแบบระบบดิจิทัลเพื่อสร้างบน FPGA พอสังเขป การอธิบายควรให้อ่านเข้าใจประเด็นต่อไปนี้ ได้แก่ กระบวนการมีขั้นตอนอะไรบ้าง ใช้เครื่องมืออะไรในการทำ ผลที่ได้ในแต่ละขั้นตอนเป็นอะไร

## VHDL Concurrent Statements

### 1. Concurrent Signal Assignment Statements

#### 1.1. Simple Concurrent Signal Assignment Statements

##### Syntax

```
<target signal> <= <waveform>;
<target signal> <= <waveform> after <time delay>;
```

#### 1.2. Conditional Signal Assignment (when-else) Statements

##### Syntax

```
<target signal> <= <waveform #1> when <condition #1> else
<waveform #2> when <condition #2> else
.
.
<waveform #(n-1)> when <condition#(n-1)> else
<waveform #n>;
```

#### 1.3. Select Signal Assignment (with-select) Statements

##### Syntax

```
with <select signal> select
<target signal> <= <waveform #1>      when <choices#1>,
<waveform #2>      when <choices#2>,
.
.
<waveform #(n-1)> when <choices#(n-1)>
<waveform #n>      when others;
```

## 2. Process Statements

### Syntax

```
[label]: process([<sensitivity list>])
[Declarations for using inside the process]
begin
<sequential statements>
end process;
```

### Possible Sequential Statements

#### 2.1. Sequential signal assignments

```
<target signal> <= <waveform>;
```

#### 2.2. Variable assignments

```
<target variable> := <value>;
```

## 2.3. If statements

```

if <condition#1> then
    <sequential statements#1>
elsif <condition#2> then
    <sequential statements#2>
    .
    .
elsif <condition#(n-1)> then
    <sequential statements#(n-1)>
else
    <sequential statements#n>
end if;

```

## 2.4. Case statements

```

case <select signal or variable> is
    when <choices#1> =>
        <sequential statements#1>
    when <choice#2> =>
        <sequential statement#2>
        .
        .
    when <choices#(n-1)> =>
        <sequential statement#(n-1)>
    when others =>
        <sequential statement#n>
end case;

```

## 2.5. Loop statements

```

for <index> in <range> loop
    <sequential statements>
end loop;

while <condition> loop
    <sequential statements>
end loop;

```

## 3. Component Instantiations

```

<label>: <component name>
    [generic map (<generic name#1> => <actual#1>,
                  <generic name#2> => <actual#2>,
                  .
                  .
                  )]
    port map (<port name#1> => <actual#1>,
              <port name#2> => <actual#2>,
              .
              .
              <port name#n> => <actual#n>);

```

## Useful VHDL Templates for describing basic building blocks

1. 2-input Multiplexor in which *z* is the output, *a0*, *a1* are the 2 inputs, *s* is the select signal

### 1.1. Using when-else

```
z <= a0 when s = '0' else
    a1;
```

### 1.2. Using with-select

```
with s select
    z <= a0 when '0',
        a1 when others;
```

2. 4-input Multiplexor in which *z* is the output, *a0*, *a1*, *a2*, *a3* are the 4 inputs, *s* is the select signal

### 2.1. Using when-else

```
z <= a0 when s = "00" else
    a1 when s = "01" else
    a2 when s = "10" else
    a3;
```

### 2.2. Using with-select

```
with s select
    z <= a0 when "00",
        a1 when "01",
        a2 when "10",
        a3 when others;
```

3. Registers using process

Register name: *r*

Clock signal : *clk*

Input signal : *r\_next*

Output signal: *r\_reg*

- 3.1. without asynchronous reset

```
process (clk)
begin
    if clk'event and clk='1' then
        r_reg <= r_next;
    end if;
end if;
```

- 3.2. with *active-high asynchronous reset* signal whose name is **reset** and the register is reset to INIT

```
process (clk, reset)
begin
    if reset='1' then
        r_reg <= INIT;
    elsif clk'event and clk='1' then
        r_reg <= r_next;
    end if;
end if;
```

## Boolean Algebra Postulates and Logic Theorems

## Huntington's Postulates of Boolean Algebra

## 1. Closure Properties

(P1a)  $a + b \in \{0, 1\}$

(P1b)  $a \cdot b \in \{0, 1\}$

## 2. Identity Properties

(P2a)  $a + 0 = a$

(P2b)  $a \cdot 1 = a$

## 3. Commutative Properties

(P3a)  $a + b = b + a$

(P3b)  $a \cdot b = b \cdot a$

## 4. Distributive Properties

(P4a)  $a + b \cdot c = (a + b) \cdot (a + c)$

(P4b)  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

## 5. Complement Properties

(P5a)  $a + \bar{a} = 1$

(P5b)  $a \cdot \bar{a} = 0$

Table 2.3 List of important logic theorems

No.	Statement	Theorem's Name
(T1a)	$X + 1 = 1$	(Null element)
(T1b)	$X \cdot 0 = 0$	
(T2a)	$X + X = X$	(Idempotency)
(T2b)	$X \cdot X = X$	
(T3)	$(X')' = X$	(Involution)
(T4a)	$(X + Y) + Z = X + (Y + Z)$	(Associativity)
(T4b)	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	
(T5a)	$X + X \cdot Y = X$	(Covering or Absorption)
(T5b)	$X \cdot (X + Y) = X$	
(T6a)	$X \cdot Y + X \cdot Y' = X$	(Combining or Adjacency)
(T6b)	$(X + Y) \cdot (X + Y') = X$	
(T7a)	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$	(Consensus)
(T7b)	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$	
(T8a)	$\overline{X + Y} = \bar{X} \cdot \bar{Y}$	(De Morgan)
(T8b)	$\overline{X \cdot Y} = \bar{X} + \bar{Y}$	