มหาวิทยาลัย
ราชภัฏนครปฐม

มหาวิทยาลัยราชภัฏนครปฐม

ยินดีต้อนรับสู่...

บทเรียนออนไลน์

รายวิชา 6562213 ไมโครโพรเซสเซอร์ 1

6562213 Microprocessor 1

โดย

ผู้ช่วยศาสตราจารย์ ธีระ กาญจนสินธุ

วท.ม.(ฟิสิกส์) วท.บ.(ฟิสิกส์)

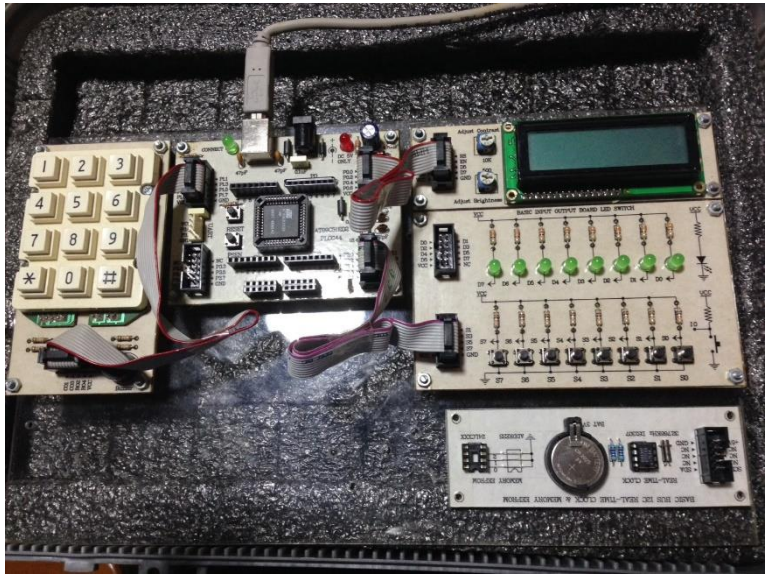
คณะวิทยาศาสตร์และเทคโนโลยี

จุดประสงค์การเรียนรู้ของรายวิชา 6562213 ไมโครโพรเซสเซอร์ 1

2) เพื่อให้ผู้เรียนสามารถ **พัฒนาซอฟต์แวร์ภาษาซี** ในการควบคุมการทำงานของฮาร์ดแวร์ที่ต่อวงจรร่วมกับชิปตระกูล MCS-51

3) เพื่อให้ผู้เรียนสามารถ **พัฒนาเจตคติและมีจรรยาบรรณ** ในการคิดค้นและพัฒนาสิ่งประดิษฐ์ โดยใช้ชิปตระกูล MCS-51 เป็นหน่วยประมวลผล

1) เพื่อให้ผู้เรียนสามารถ **ออกแบบและพัฒนาฮาร์ดแวร์** ที่มีไมโครคอนโทรลเลอร์ หรือชิปตระกูล MCS-51 เป็นหน่วยประมวลผล



หัวข้อสำคัญของรายวิชา 6562213 ไมโครโพรเซสเซอร์ 1 บทที่ 1-6

- บทที่ 1 สถาปัตยกรรมไมโครโพรเซสเซอร์และไมโครคอนโทรลเลอร์พื้นฐาน
- บทที่ 2 โครงสร้างภายในของไมโครคอนโทรลเลอร์
- บทที่ 3 ชุดคำสั่งแยกตามประเภทการใช้งาน
- บทที่ 4 ตัวอย่างการเขียนโปรแกรมเบื้องต้นและโปรแกรมใช้งาน
- บทที่ 5 การออกแบบวงจรหน่วยความจำและอุปกรณ์ต่อร่วม
- บทที่ 6 การประยุกต์ใช้งานไมโครคอนโทรลเลอร์

มหาวิทยาลัย
ราชภัฏนครปฐม

รายวิชา 6562213 ไมโครโพรเซสเซอร์ 1

บทที่ 3 ชุดคำสั่งแยกตามประเภทการใช้งาน

โดยผู้ช่วยศาสตราจารย์ธีระ กาญจนสินธุ

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

บทนำ

ในบทนี้ มีเนื้อหาเกี่ยวกับ โครงสร้างของโปรแกรมภาษาซี ชนิดตัวแปรและการตั้งชื่อตัวแปรในภาษาซี (identifier) ตัวแปรชุด (Array Variable) ชุดคำสั่งเกี่ยวกับการกำหนดค่า ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ ชุดคำสั่งเกี่ยวกับเงื่อนไข (if-condition, switch case condition)

การนำเสนอตัวอย่างประกอบเพื่อสร้างความเข้าใจการใช้รูปแบบ มีกรณีศึกษาที่พัฒนาความเข้าใจได้หลากหลาย ผู้เรียนจึงจำเป็นต้องสังเกตความแตกต่างและปรับความเข้าใจไปในแต่ละกรณี

เนื้อหาในบทนี้ มีส่วนสำคัญในการพัฒนาโปรแกรมควบคุมต่อไป

หัวข้อสำคัญในบทที่ 3

3.1 ทบทวนโครงสร้างของโปรแกรมภาษา ซี

3.2 ทบทวน ชนิดตัวแปรและการตั้งชื่อตัวแปรในภาษาซี (Identifier)

3.3 ทบทวนตัวแปรชุด (Array Variable)

3.4 ชุดคำสั่งเกี่ยวกับการกำหนดค่า

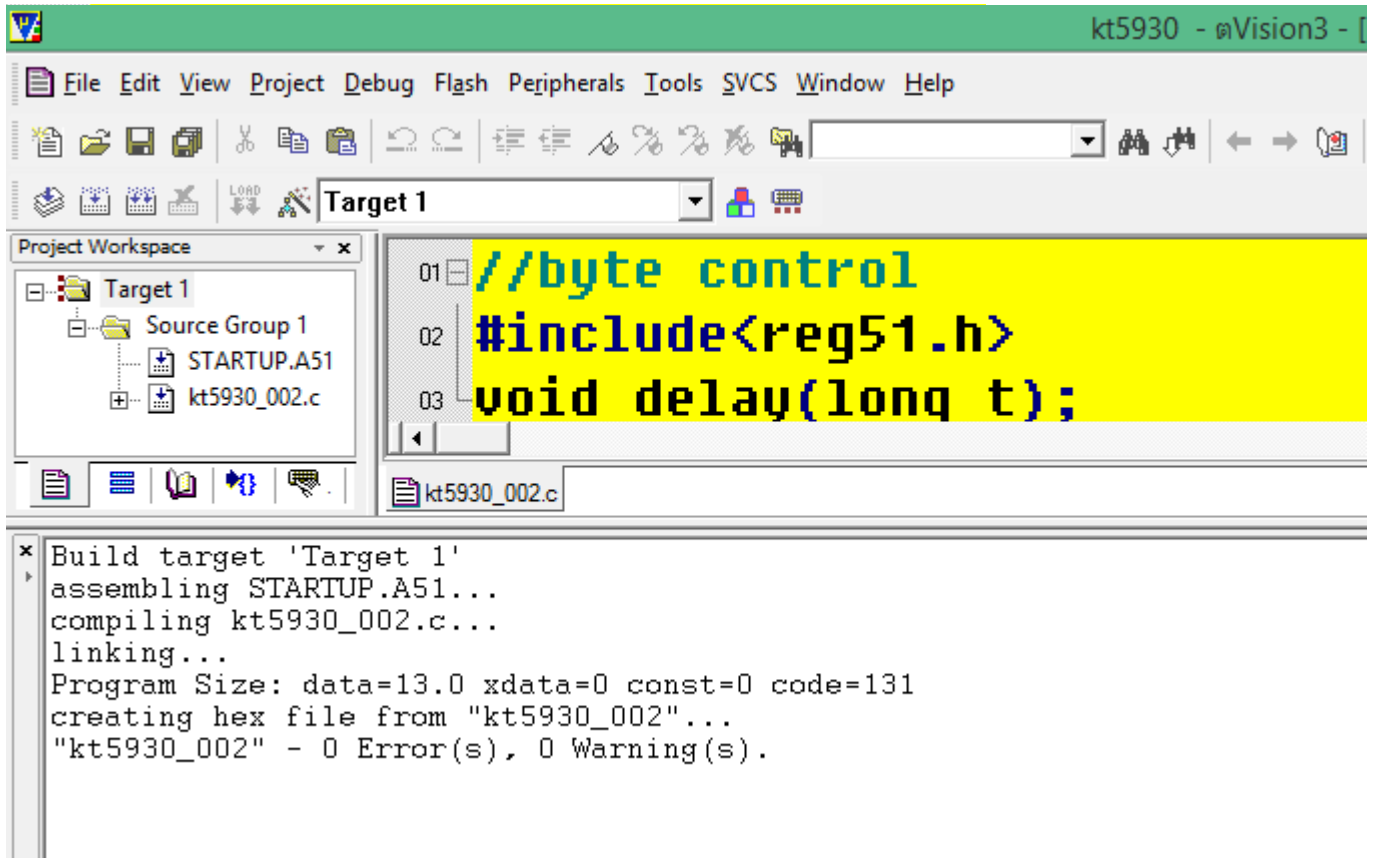
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

3.6 ชุดคำสั่งเกี่ยวกับเงื่อนไข (condition)

3.1 ทบทวน โครงสร้างโปรแกรมภาษาซี

นิยาม | C - Editor | C - Compiler | Interpreter

C editor
พื้นที่เขียนภาษาซี ซึ่งอาจเป็น Notepad/ Text Editor หนึ่งใดและเขียนตามรูปแบบมาตรฐานของภาษา



C compiler
ซอฟต์แวร์ของผู้ผลิตชิป
ซึ่งช่วยให้ผู้พัฒนานวัตกรรม
แปลเปลี่ยนภาษาซี เป็นภาษาเครื่อง
(จากไฟล์สกุล .C เป็นไฟล์สกุล .hex)
โดยแปลทั้งหมดในครั้งเดียว รวดเร็ว

Interpreter แปลทีละบรรทัด ช้ากว่า

3.1 ทบทวน โครงสร้างภาษาซี

Preprocessing Directives | Main Function | User Defined Function | Comment

Preprocessing Directives
ส่วนหัวของภาษาซี ที่มักขึ้นต้นด้วย # โดยเป็นที่รวมของไลบารี่(header file)ที่ใช้เช่น #include #define #if

Main Function (Main=หลัก)
คือฟังก์ชันหลักของโปรแกรมสกุล .C main function อยู่ในส่วนของ {.....}

User Defined Function
คือฟังก์ชันที่ถูกเรียกจาก main() หรือจากฟังก์ชันอื่นภายในโปรแกรม main function ในส่วนของ {.....}

Comment ส่วนที่อธิบายโปรแกรม

Line no.

```
/*.....*/                          ..Comment
#include<reg51.h>                  ..Preprocessing Directives
Void xxx(.....);                   ..User Defined Function
void main()                        ..Main Function
{P1=0x7F;delay(50000);
 P1=0xBF;delay(50000);
 P1=0x3F;delay(50000);
 P1=0xFF;delay(50000);
 }

 Void xxx(.....)                   ..User Defined Function
void delay(long t)
{.....}
{for (t=t;t>0;t--);}
```

3.1 ทบทวน โครงสร้างภาษาซี

Preprocessing Directives | Main Function | User Defined Function | Comment

Preprocessing Directives
ส่วนหัวของภาษาซี ที่มักขึ้นต้นด้วย # โดยเป็นที่รวมของไลบารี่(header file)ที่ใช้เช่น #include #define #if

Main Function (Main=หลัก)
คือฟังก์ชันหลักของโปรแกรมสกุล .C main function อยู่ในส่วนของ {.....}

User Defined Function
คือฟังก์ชันที่ถูกเรียกจาก main() หรือจากฟังก์ชันอื่นภายในโปรแกรม main function ในส่วนของ {.....}

Comment ส่วนที่อธิบายโปรแกรม

```
Line no.
1   /*.....*/                    ..Comment
2   #.....                       ..Preprocessing Directives
3   Void xxx(.....);             ..User Defined Function
4
5   Void main()                  ..Main Function
6   {
7
8   ..................
9
10  ..................
11  }
12  Void xxx(.....)              ..User Defined Function
13
14  {.....}
```

...อาจจัดโครงสร้างอีกลักษณะหนึ่ง ดังภาพ

3.1 ทบทวน โครงสร้างภาษาซี

Preprocessing Directives | Main Function | User Defined Function | Comment

Preprocessing Directives
ส่วนหัวของภาษาซี ที่มักขึ้นต้นด้วย # โดยเป็นที่รวมของไลบารี่(header file)ที่ใช้เช่น #include #define #if

Main Function (Main=หลัก)
คือฟังก์ชันหลักของโปรแกรมสกุล .C main function อยู่ในส่วนของ {.....}

User Defined Function
คือฟังก์ชันที่ถูกเรียกจาก main() หรือจากฟังก์ชันอื่นภายในโปรแกรม main function ในส่วนของ {.....}

Comment ส่วนที่อธิบายโปรแกรม

```
01 //byte control                                   ..Comment
02 #include<reg51.h>                               ..Preprocessing Directives
03 void delay(long t);                             ..User Defined Function
04 void main()                                     ..Main Function
05 {P1=0x7F;delay(50000);
06  P1=0xBF;delay(50000);
07  P1=0x3F;delay(50000);
08  P1=0xFF;delay(50000);
09 }
10
11 void delay(long t)                              ..User Defined Function
12 {for (t=t;t>0;t--);}
```

3.2 ทบทวน การตั้งชื่อตัวแปรในภาษาซี (Identifier)

นิยาม **Identifier** คือการตั้งชื่อตัวแปรเพื่อแทนข้อมูล หรือฟังก์ชัน หรือลาเบล

```
01 //byte control
02 #include<reg51.h>
03 void delay(long t);
04 void main()
05 {P1=0x7F;delay(50000);
06  P1=0xBF;delay(50000);
07  P1=0x3F;delay(50000);
08  P1=0xFF;delay(50000);
09  }
10
11 void delay(long t)
12 {for (t=t;t>0;t--);}
```

ตัวแปรแทนข้อมูล ชื่อ **t**

ตัวแปรแทนฟังก์ชัน ชื่อ **delay**

3.2 ทบทวน การตั้งชื่อตัวแปรในภาษาซี (Identifier)

นิยาม **Identifier** คือการตั้งชื่อตัวแปรเพื่อแทนข้อมูล หรือฟังก์ชัน หรือลาเบล

```
01 //byte control
02 #include<reg51.h>
03 void delay(long t);
04 void main()
05 {P1=0x7F;delay(50000);
06
07
08
09
10
11 void delay(long t)
12 {for(t=t;t>0;t--);}
```

ตัวแปรแทนข้อมูล ชื่อ t

ตัวแปรแทนฟังก์ชัน ชื่อ **delay**

กฎเกณฑ์การตั้งชื่อตัวแปร

1) ชื่อตัวแปรต้องไม่ซ้ำกับ reserve word

asm	auto			goto	if	int	long	near
break				return	register	short	signed	
case	char	const	continue			struct	sizeof	switch
default	do	double		static	typedef			
else	enum	extry	extern			union	unsigned	
far	float	for		void	volatile	while		

3.2 ทบทวน การตั้งชื่อตัวแปรในภาษาซี (Identifier)

นิยาม **Identifier** คือการตั้งชื่อตัวแปรเพื่อแทนข้อมูล หรือฟังก์ชัน หรือลาเบล

```
01 //byte control
02 #include<reg51.h>
03 void delay(long t);
04 void main()
05 {P1=0x7F;delay(50000);
06
07
08
09
10
11 void delay(long t)
12 {for(t=t;t>0;t--);}
```

ตัวแปรแทนข้อมูล ชื่อ t

ตัวแปรแทนฟังก์ชัน ชื่อ **delay**

กฎเกณฑ์การตั้งชื่อตัวแปร

1) ชื่อตัวแปรต้องไม่ซ้ำกับ reserve word

2) ชื่อตัวแปรเป็น case sensitive
เช่น ตัวแปรชื่อ Ab/aB/AB/ab

Ab aB AB ab

3) ชื่อตัวแปร**ต้องนำด้วยตัวอักษร** 1ตัว ตามด้วย**ตัวเลข** และ/หรือ**ขีดล่าง** _ เท่านั้นโดยไม่มีช่องว่าง เช่น
ตัวแปรชื่อ
A1_a/ a_B2/ g1_M1_s00

A_B ab2 a1b2 M1_s0 M1_s0g1

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

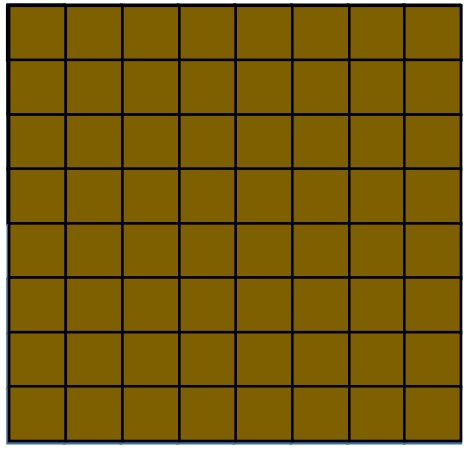
```
01 //byte control
02 #include<reg51.h>
03 void delay(long t);
04 void main()
05 {P1=0x7F;delay(50000)
06  P1=0xBF;delay(50000)
08  P1=0xFF;delay(50000)
09 }
10
11 void delay(long t)
12 {for (t=t;t>0;t--);}
```

ชนิดของตัวแปร t คือ long

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
sbit	1 bit	0	1
char	8 bits	-128	+127
Unsigned char	8 bits	0	255
int	16 bits	-32,768	+32,767
Unsigned int	16 bits	0	65,535
long	32 bits	-2,147,483,648	+2,147,483,647
Unsigned long	32 bits	0	4,294,967,295
float	32 bits	3.4×10^{-38}	$3.4 \times 10^{+38}$
double	64 bits	1.7×10^{-308}	$1.7 \times 10^{+308}$

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว



Memory area
8 bits

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
sbit	1 bit	0	1
char	8 bits	-128	+127
Unsigned char	8 bits	0	255
int	16 bits	-32,768	+32,767
Unsigned int	16 bits	0	65,535
long	32 bits	-2,147,483,648	+2,147,483,647
Unsigned long	32 bits	0	4,294,967,295
float	32 bits	3.4×10^{-38}	$3.4 \times 10^{+38}$
double	64 bits	1.7×10^{-308}	$1.7 \times 10^{+308}$

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
sbit	1 bit	0	1

ตัวอย่าง กำหนดตัวแปรชนิด sbit

```
01 //bit control
02 #include<reg51.h>
03 sbit l0=P0^0;
04 void delay(int t);
05
06 void main()
07 {l0=0;delay(10000);
08  l0=1;delay(20000);
09 }
10 void delay(int t)
11 {for (t=t;t>0;t--);}
```

```
Sbit l0=P0^0;
```

```
/* กำหนดให้ตัวแปร l0 (แอลศูนย์) เป็นตัวแปรชนิด sbit
   โดยแทนข้อมูลของบิต P0^0 (P0.0)    */
/* กำหนดให้ตัวแปร l0 = 0 (แอลศูนย์เท่ากับศูนย์)      */
/* กำหนดให้ตัวแปร l0 = 1 (แอลศูนย์เท่ากับหนึ่ง)      */
```

ตัวแปรชนิด sbit :
นิยมใช้งานแทน SFR (เรจิสเตอร์หน้าที่พิเศษ) 80h-FFh
และ/หรือ Bit addressable (ข้อมูลที่กำหนดระดับบิต) 20h-2Fh

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
Unsigned char	8 bit	0	255

ตัวอย่าง กำหนดตัวแปรชนิด unsigned char

```
01 //bit control_unsigned char
02 #include<reg51.h>
03 sbit l0=P0^0;
04 void delay(unsigned char t);
05
06 void main()
07 {l0=0;delay(250);
08  l0=1;delay(250);
09 }
10 void delay(unsigned char t)
11 {for (t=t;t>0;t--);}
```

Delay (unsigned char t);

/* กำหนดให้ตัวแปร t (ที) เป็นตัวแปรชนิด **unsigned char** โดย t มีค่าตั้งแต่ **0** ถึง **255** ในฟังก์ชัน */

/* กำหนดให้ตัวแปร t = (ค่าตั้งต้น) */
/* หาก t มากกว่า 0 (ให้ลดค่าลง หนึ่ง) */

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
Unsigned int	16 bit	0	65535

ตัวอย่าง กำหนดตัวแปรชนิด unsigned int

```
01 //bit control_unsigned int
02 #include<reg51.h>
03 sbit l0=P0^0;
04 void delay(unsigned int t);
05
06 void main()
07 {l0=0;delay(65000);
08  l0=1;delay(65000);
09 }
10 void delay(unsigned int t)
11 {for (t=t;t>0;t--);}
```

Delay (unsigned int t);

/* กำหนดให้ตัวแปร t (ที) เป็นตัวแปรชนิด **unsigned integer** โดย t มีค่าตั้งแต่ **0** ถึง **65,535** ในฟังก์ชัน */

/* กำหนดให้ตัวแปร t = (ค่าตั้งต้น) */
/* หาก t มากกว่า 0 (ให้ลดค่าลง หนึ่ง) */

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
Unsigned long	32 bit	0	4,294,967,295

ตัวอย่าง กำหนดตัวแปรชนิด unsigned long

```
01 //bit control_unsigned long
02 #include<reg51.h>
03 sbit l0=P0^0;
04 void delay(unsigned long t);
05
06 void main()
07 {l0=0;delay(4200000);
08  l0=1;delay(4200000);
09 }
10 void delay(unsigned long t)
11 {for (t=t;t>0;t--);}
```

Delay (unsigned long t);

/* กำหนดให้ตัวแปร t (ที) เป็นตัวแปรชนิด **unsigned long** โดย t มีค่าตั้งแต่ **0** ถึง **4,294,967,295** */

/* กำหนดให้ตัวแปร t = (ค่าตั้งต้น) */
/* หาก t มากกว่า 0 (ให้ลดค่าลง หนึ่ง) */

3.2 ทบทวน ชนิดของข้อมูลในภาษาซี (Data Type)

นิยาม **ชนิดของตัวแปร** คือ ตัวระบุขนาดหน่วยความจำที่เก็บตัวแปร 1 ตัว

Sbit → bit

Data type	size	ค่าต่ำสุด	ค่าสูงสุด
sbit	1 bit	0	1
Unsigned char	8 bits	0	255
Unsigned int	16 bits	0	65,535
Unsigned long	32 bits	0	4,294,967,295

```
//bit control_unsigned char
#include<reg51.h>
sbit l0=P0^0;
void delay(unsigned char t);

void main()
{l0=0;delay(250);
 l0=1;delay(250);
}
void delay(unsigned char t)
{for (t=t;t>0;t--);}
```

```
//bit control_unsigned int
#include<reg51.h>
sbit l0=P0^0;
void delay(unsigned int t);

void main()
{l0=0;delay(65000);
 l0=1;delay(65000);
}
void delay(unsigned int t)
{for (t=t;t>0;t--);}
```

```
//bit control_unsigned long
#include<reg51.h>
sbit l0=P0^0;
void delay(unsigned long t);

void main()
{l0=0;delay(4200000);
 l0=1;delay(4200000);
}
void delay(unsigned long t)
{for (t=t;t>0;t--);}
```

3.3 ทบทวนตัวแปรชุด (Array Variable)

นิยาม ตัวแปรชุด คือตัวแปรชื่อเดียวที่ใช้เลขในวงเล็บกำกับเก็บค่าเป็นชุด

ตัวอย่างประกาศตัวแปรชุด 1 มิติ เป็นตัวแปรที่เก็บข้อมูลชุดแบบแถวเดียว

```
Unsigned int xt[4]={0x00,0x81,0xC3,0xE7,0xFF}
```

0x00	0x81	0xC3	0xE7	0xFF

```
Unsigned char segm[4]={0x3F,0x06,0x5B,0x4F,0x66}
```

0x3F	0x06	0x5B	0x4F	0x66

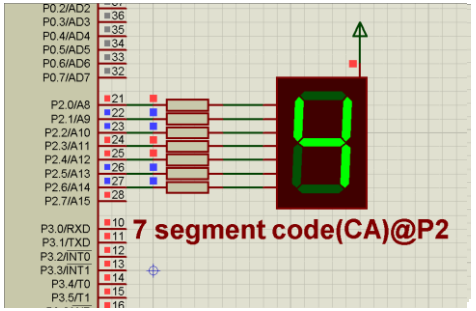
ตัวอย่างประกาศตัวแปรชุด 2 มิติ เป็นตัวแปรที่เก็บข้อมูลชุดแบบตารางเดียวกัน

```
Unsigned char x [2][4]={0x00,0x81,0xC3,0xE7,0xFF, 0x3F,0x06,0x5B,0x4F,0x66}
```

0x3F	0x06	0x5B	0x4F	0x66
0x00	0x81	0xC3	0xE7	0xFF

3.3 ทบทวนตัวแปรชุด (Array Variable)

ตัวอย่างโปรแกรมที่มีการประกาศและการใช้ตัวแปรชุด 1 มิติ




```
//@P2 display 0 1 2 3 4
#include<reg51.h>
char segm[]={0xC0,0xF9,0xA4,0xB0,0x99};   //ตัวแปรที่เก็บข้อมูลชุดแบบแถวเดียว
unsigned int i;
void delay(unsigned int t);
void main()
{i=0;
 while(1)
 {P2=segm[i];delay();i++;                 //ส่งค่าข้อมูลจากที่เก็บอ้างอิงตามตัวเลขกำกับออกพอร์ต P2
  if(i>4){i=0;}; }
}
void delay(unsigned int t)
{ for(t=0;t<4;t++);}
```

0xC0	0xF9	0xA4	0xB0	0x99

Unsigned char segm[4]={0xC0,0xF9,0xA4,0xB0,0x99}

0xC0	0xF9	0xA4	0xB0	0x99

3.4 ชุดคำสั่งเกี่ยวกับการกำหนดค่า

ตัวอย่างการกำหนดค่า : bit sbit unsigned int Array variableds

```
#include<reg51.h>
bit led;
led=1;
```

กำหนดตัวแปรชนิด bit ชื่อ led
กำหนดให้ตัวแปร led เท่ากับ 1

bit :ใช้กับข้อมูล 1 บิต

```
#include<reg51.h>
sbit P10=P1^0;
P10=0;
```

กำหนดตัวแปรชนิด sbit ชื่อ P10
กำหนดตัวแปร P10 เท่ากับ 0
กรณีนี้ ค่า P10 เปลี่ยนจะส่งผลที่ P1.0

sbit :ใช้กับ SFR register/ (ข้อมูลที่เข้าถึงได้ระดับบิต 20-2Fh) sbit นิยมใช้แสดงสภาวะ SFR ระดับบิต

```
#include<reg51.h>
unsigned int x;
x=0xFE;
```

กำหนดตัวแปรชนิด unsigned int ชื่อ x
กำหนดตัวแปร x เท่ากับ (มีค่าได้ตั้งแต่ 0-65535)
กรณีนี้ ตัวแปร x เป็นข้อมูลขนาด 16 บิต

Unsigned int :ใช้กำหนดให้ตัวแปร (มีค่า0-65535/00h-FFFFh) ที่เป็นข้อมูล 16 บิต

```
#include<reg51.h>
char x[]={0xC0,0xF9,0xA4};
```

กำหนดตัวแปรชนิด char ชื่อ x
กำหนดค่า x[0]=0xC0, x[1]=0xF9, x[2]=0xA4
กรณีนี้ ตัวแปร x เป็นตัวแปรชุดขนาด 8 บิต

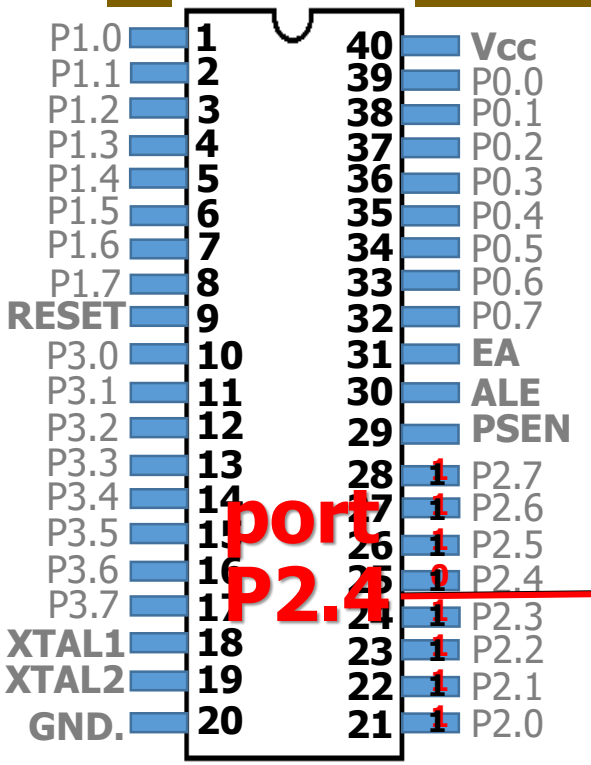
array :ใช้ชื่อตัวแปรเดียวกันโดยมีสมาชิกย่อยที่ชี้ระบุเลขตำแหน่ง (index)

3.4 ชุดคำสั่งเกี่ยวกับการกำหนดค่า

รูปแบบวนทำซ้ำ

```
while(1)
{....}
```
วนทำซ้ำ

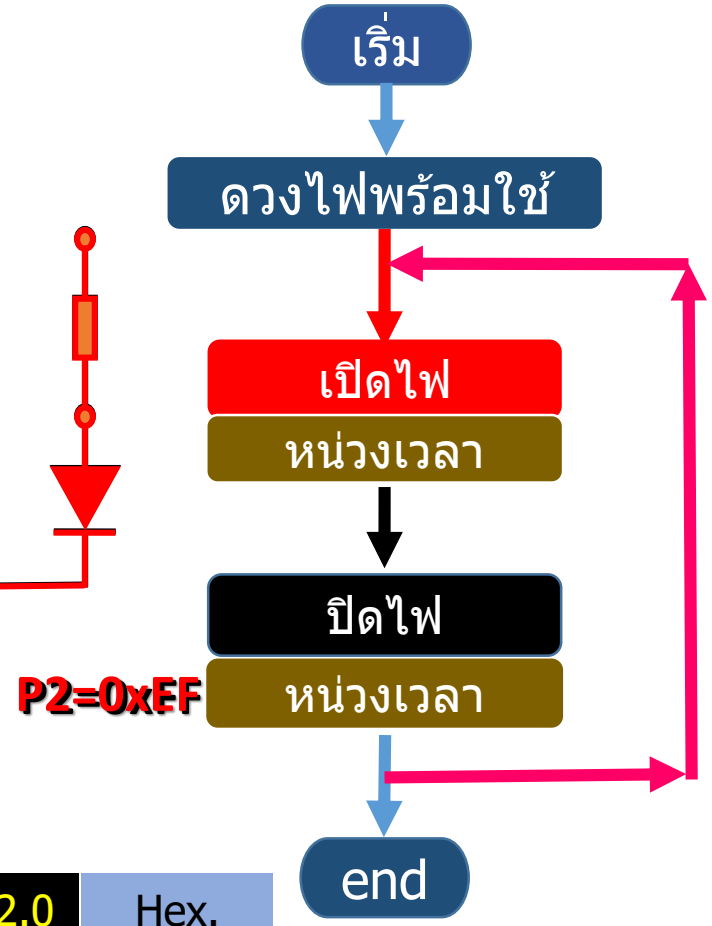
ตัวอย่างที่ใช้เอาต์พุต



เปิด/ปิด ดวงไฟ 1 ดวง

P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	Hex.
1	1	1	0	1	1	1	1	0xEF

ผังความคิด (flowchart)



วงจรดวงไฟใช้ port P2.4

```
#include<reg51.h>
void delay (long t);
void main ()
{ P2=0xFF;
 while(1)
{P2=0xEF; delay(100);
  P2=0xFF; delay(100);}
 }
void delay (long t)
 {for (t=t;t>0;t--);}
```

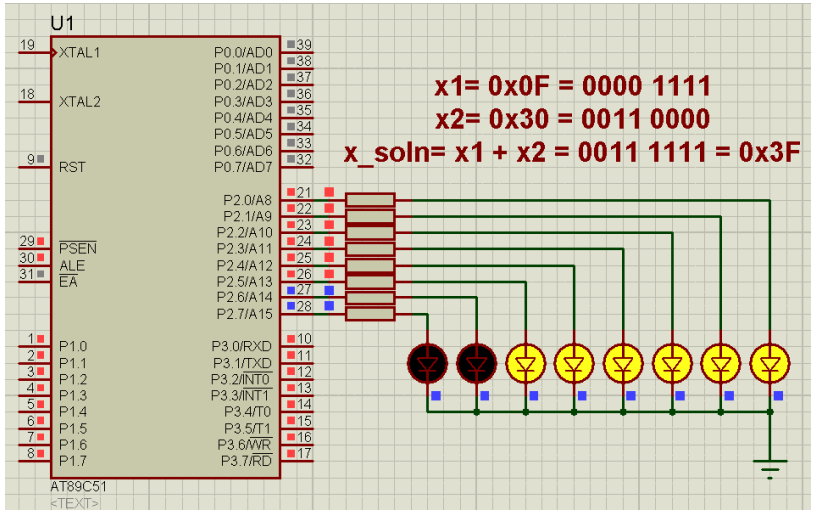
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลขคณิต: + -

+: การบวกระหว่างข้อมูลขนาด 8 บิต (x1+x2 =x_soln)

X1=0x0Fh=0000 1111b
X2=0x30h=0011 0000b +
x_soln =0x3Fh=0011 1111b

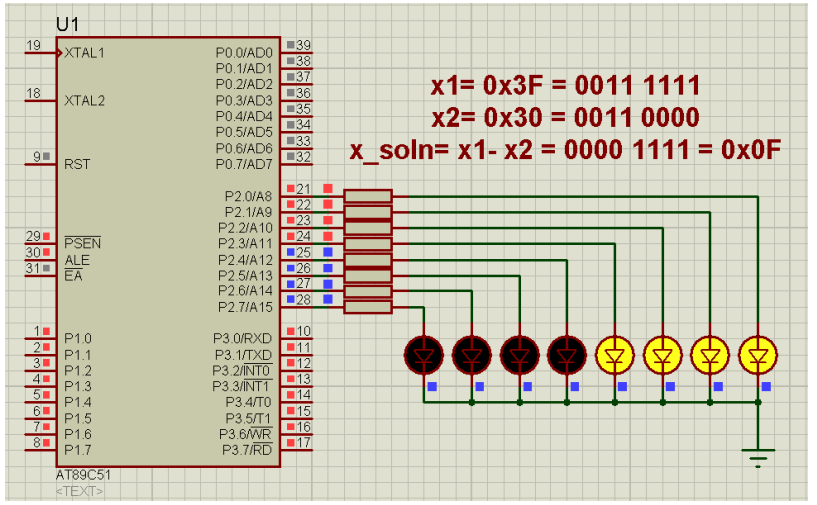
```
#include<reg51.h>
int x1,x2,x_soln;
void main()
{x1=0x0F; x2=0x30;
 x_soln=x1+x2;
 P2=x_soln;}
```



-: การลบระหว่างข้อมูลขนาด 8 บิต (x1-x2 =x_soln)

X1=0x3Fh=0011 1111b
X2=0x30h=0011 0000b -
x_soln =0x0Fh=0000 1111b

```
#include<reg51.h>
int x1,x2,x_soln;
void main()
{x1=0x3F; x2=0x30;
 x_soln=x1-x2;
 P2=x_soln;}
```



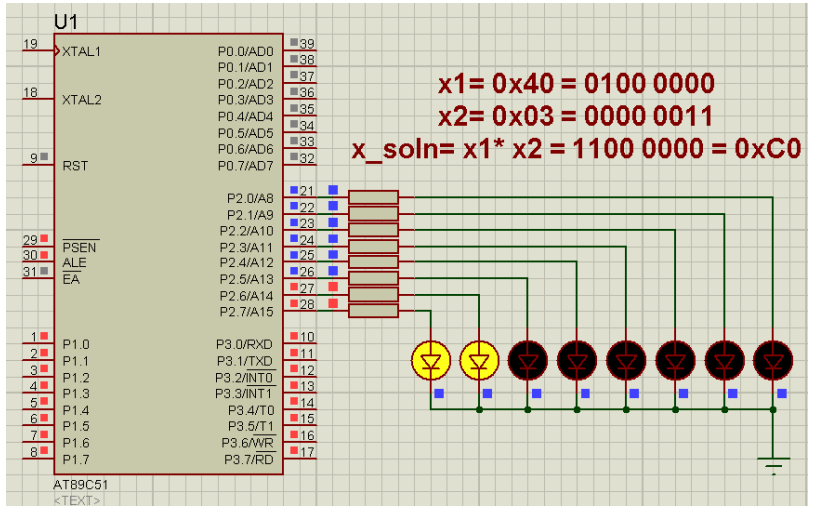
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลขคณิต: + - * /

× : การคูณระหว่างข้อมูลขนาด 8 บิต (x1*x2 =x_soln)

X1=0x40h=0100 0000b
X2=0x03h=0000 0011b
x_soln =0xC0h=1100 0000b

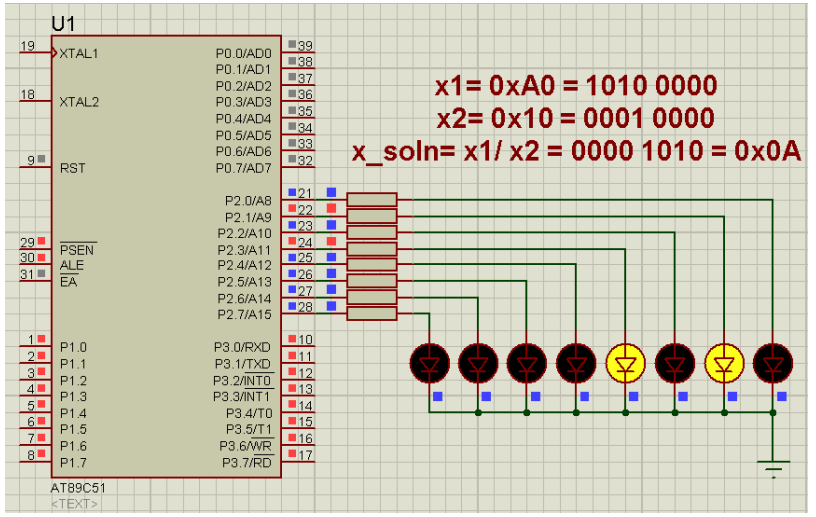
```
#include<reg51.h>
int x1,x2,x_soln;
void main()
{x1=0x40; x2=0x03;
 x_soln=x1*x2;
 P2=x_soln;}
```



÷ : การหารระหว่างข้อมูลขนาด 8 บิต (x1/x2 =x_soln)

X1=0xA0h=1010 0000b
X2=0x10h=0001 0000b
x_soln =0x0Ah=0000 1010b

```
#include<reg51.h>
int x1,x2,x_soln;
void main()
{x1=0xA0; x2=0x10;
 x_soln=x1/x2;
 P2=x_soln;}
```



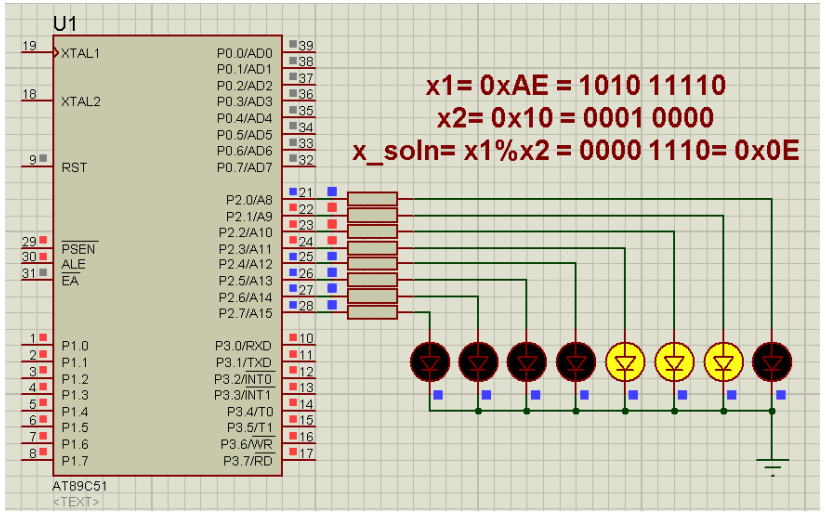
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลขคณิต: + - * / % ++

%: การหาร(เอาเศษ)ระหว่างข้อมูลขนาด 8 บิต (5%3=2)

X1=0xAEh=10101 1110b
X2=0x10h=0001 0000b **%**
x_soln =0x0Eh=0000 1110b

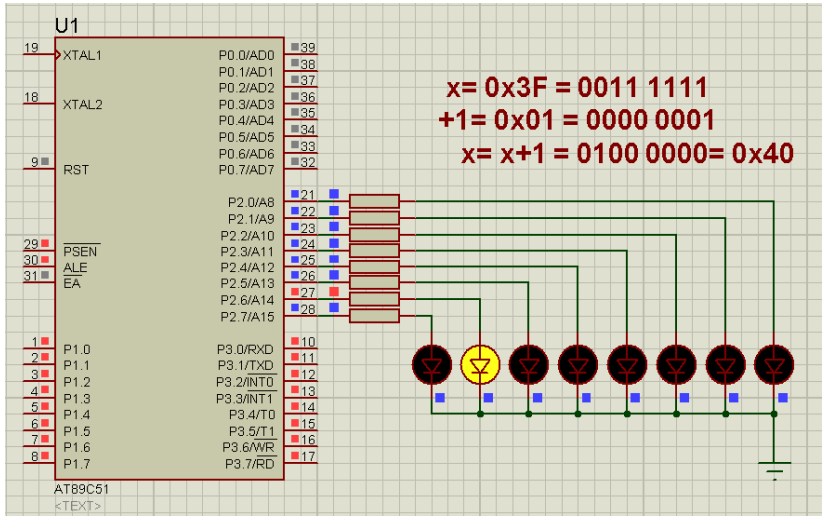
```
#include<reg51.h>
int x1,x2,x_soln;
void main()
{x1=0xAE; x2=0x10;
 x_soln=x1%x2;
 P2=x_soln;}
```



++: การเพิ่มค่าข้อมูลขนาด 8 บิตอีก 1 [x++ means (x=x+1)]

x=0x3Fh=0011 1111b
x+1=0x3Fh+1=0100 0000b **x++**
x_soln =0x40h=0100 0000b

```
#include<reg51.h>
unsigned int x;
void main()
{x=0x3F;
 x++;
 P2=x;}
```



3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

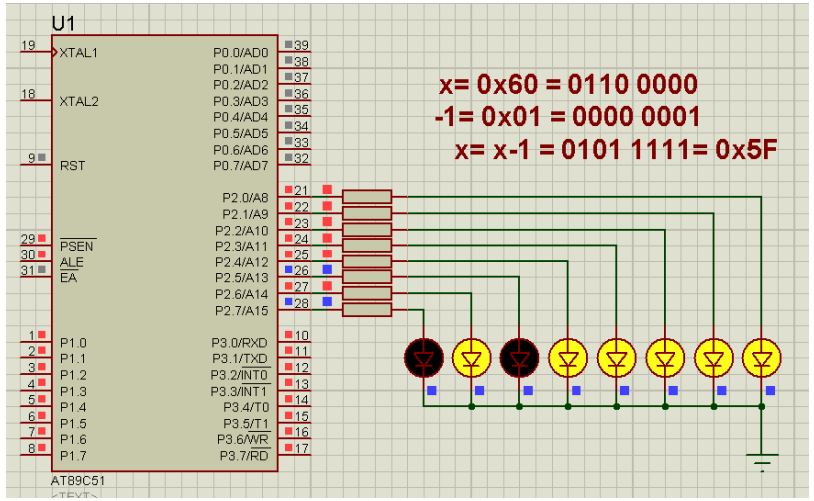
ตัวอย่างเกี่ยวกับเลขคณิต: + - * / % ++ -- +=

--: การลดค่าข้อมูลขนาด 8 บิตอีก 1 [x-- means (x=x-1)]

x=0x60h=0110 0000b
x-1 = 0x60h-1=0101 1111b
x_soln =0x5Fh=0101 1111b

x--

```
#include<reg51.h>
unsigned int x;
void main()
{x=0x60;
 x--;
 P2=x;}
```

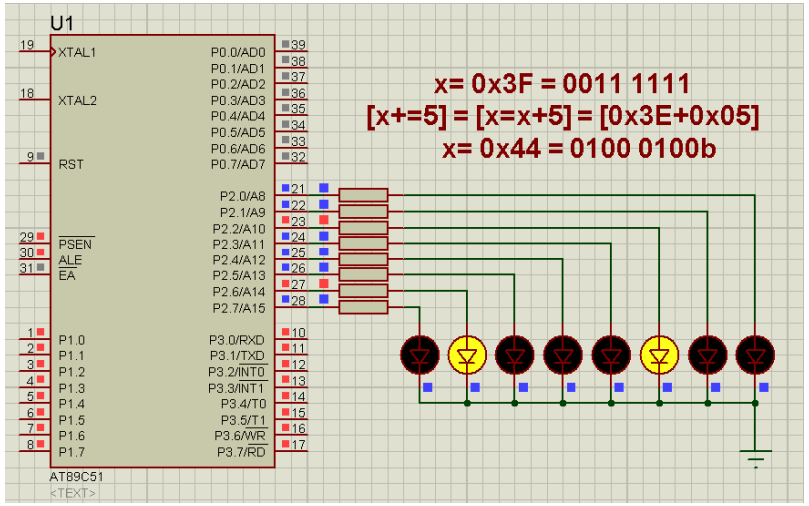


+=: การบวกเพิ่มอีกด้วยค่าทางขวามือ[x+=5 means (x=x+5)]

X=0x3Fh=0011 1111b
[X+=5]=[x=x+5]
=0x3Fh+5
=0x44h
=0100 0100b
x_soln =0x44h=0100 0100b

X+=5

```
#include<reg51.h>
unsigned int x;
void main()
{x=0x3F;
 x+=5;
 P2=x;}
```



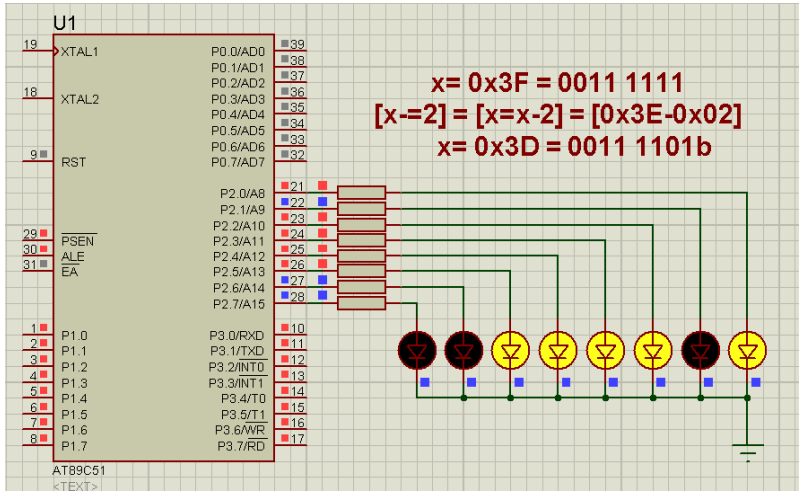
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลขคณิต: + - * / % ++ -- += -= *=

-=: การลดค่าลงอีกด้วยค่าทางขวามือ[x-=2 means (x=x-2)]

X=0x3Fh=0011 1111b
[X-=2]=[x=x-2] **X-=2**
=0x3Fh-2
=0x3Dh
=0011 1101b
x_soln =0x3Dh=0011 1101b

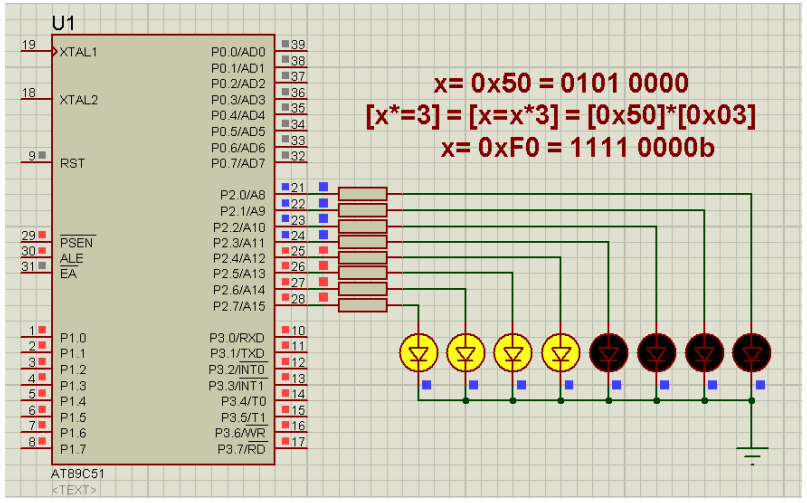
```
#include<reg51.h>
unsigned int x;
void main()
{x=0x3F;
 x-=2;
 P2=x;}
```



=: การคูณด้วยค่าทางขวามือ[x=3 means (x=x*3)]

X=0x50h=0101 0000b
[X*=3]=[x=x*3] **x*=3**
=(0x50h)*(0x03h)
x_soln =0xF0h=1111 0000b

```
#include<reg51.h>
unsigned int x;
void main()
{x=0x50;
 x*=3;
 P2=x;}
```



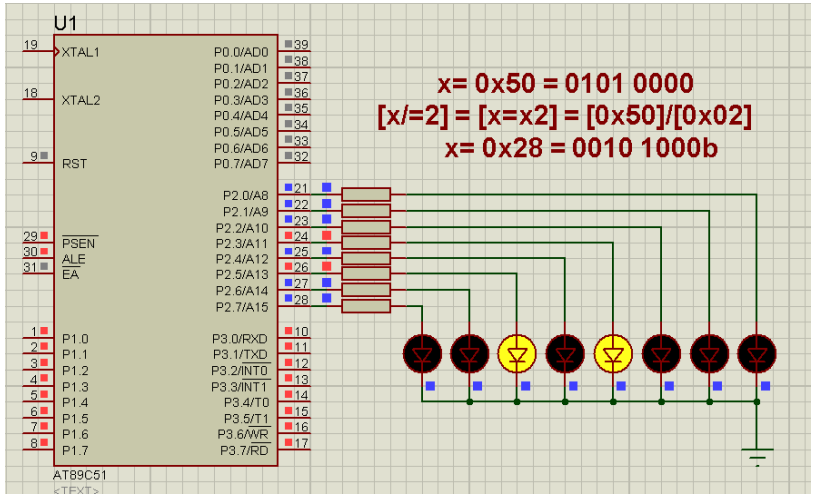
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลขคณิต: + - * / % ++ -- += -= *= ÷=

÷=: การหารด้วยค่าทางขวามือ[x/=2 means (x=x/2)]

X=0x50h=0101 0000b
[X/=2]=[x=x/2]
=(0x50h)/(0x02h)
x_soln =0x28h=0010 1000b

```
1  #include<reg51.h>
2  unsigned int x;
3  void main()
4  {x=0x50;
5   x/=2;
6   P2=x;}
7
```



3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับการเปรียบเทียบ: > >= < <= == !=

ตัวอย่างเกี่ยวกับตรรกศาสตร์: ~ & | ^

ตัวอย่างเกี่ยวกับเลื่อนบิต >> <<

>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ
==	เท่ากับ
!=	ไม่เท่ากับ

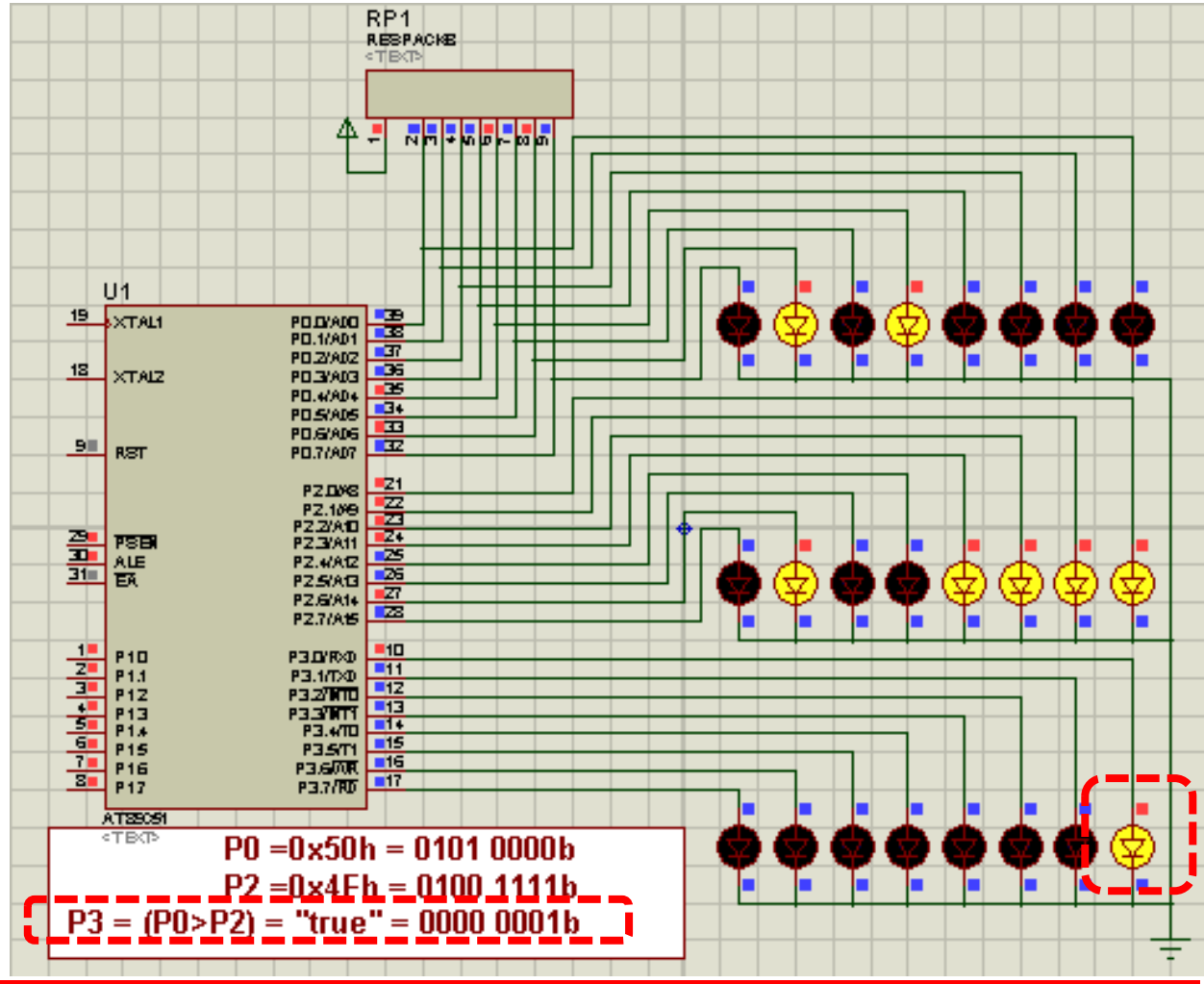
Byte operation	
~	นิเสธ(NOT)
&	และ (AND operation)
|	หรือ (OR operation)

>>	เลื่อนบิตไปทางขวา
<<	เลื่อนบิตไปทางซ้าย

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับการเปรียบเทียบ: > >= < <= == !=

```
#include<reg51.h>
void main()
{  P0=0x50;
   P2=0x4F;
   P3=(P0>P2);  }
```

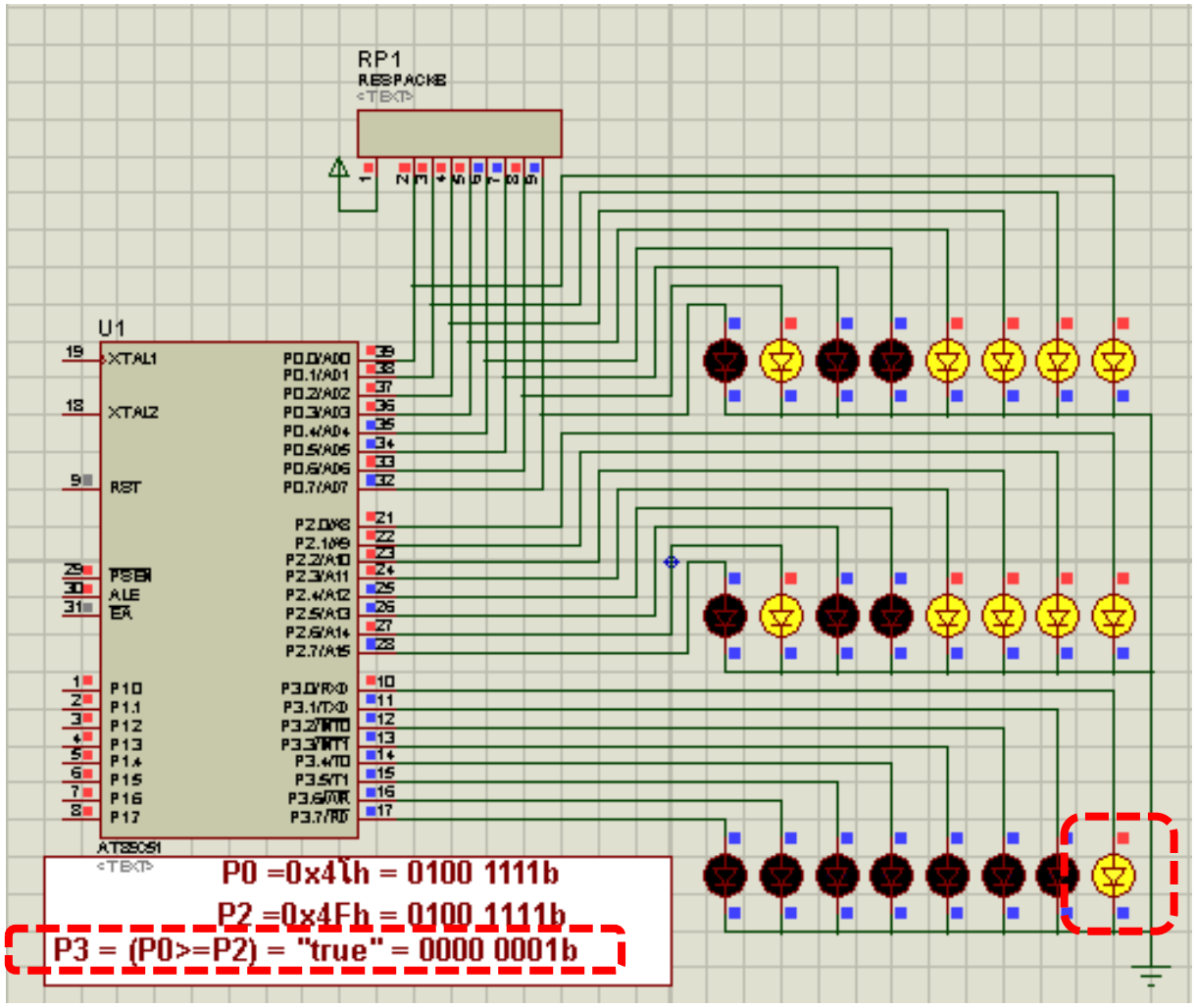


>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ
==	เท่ากับ
!=	ไม่เท่ากับ

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับการเปรียบเทียบ: > >= < <= == !=

```
1  #include<reg51.h>
2  void main()
3  { P0=0x4F;
4    P2=0x4F;
5    P3=(P0>=P2); }
6
```

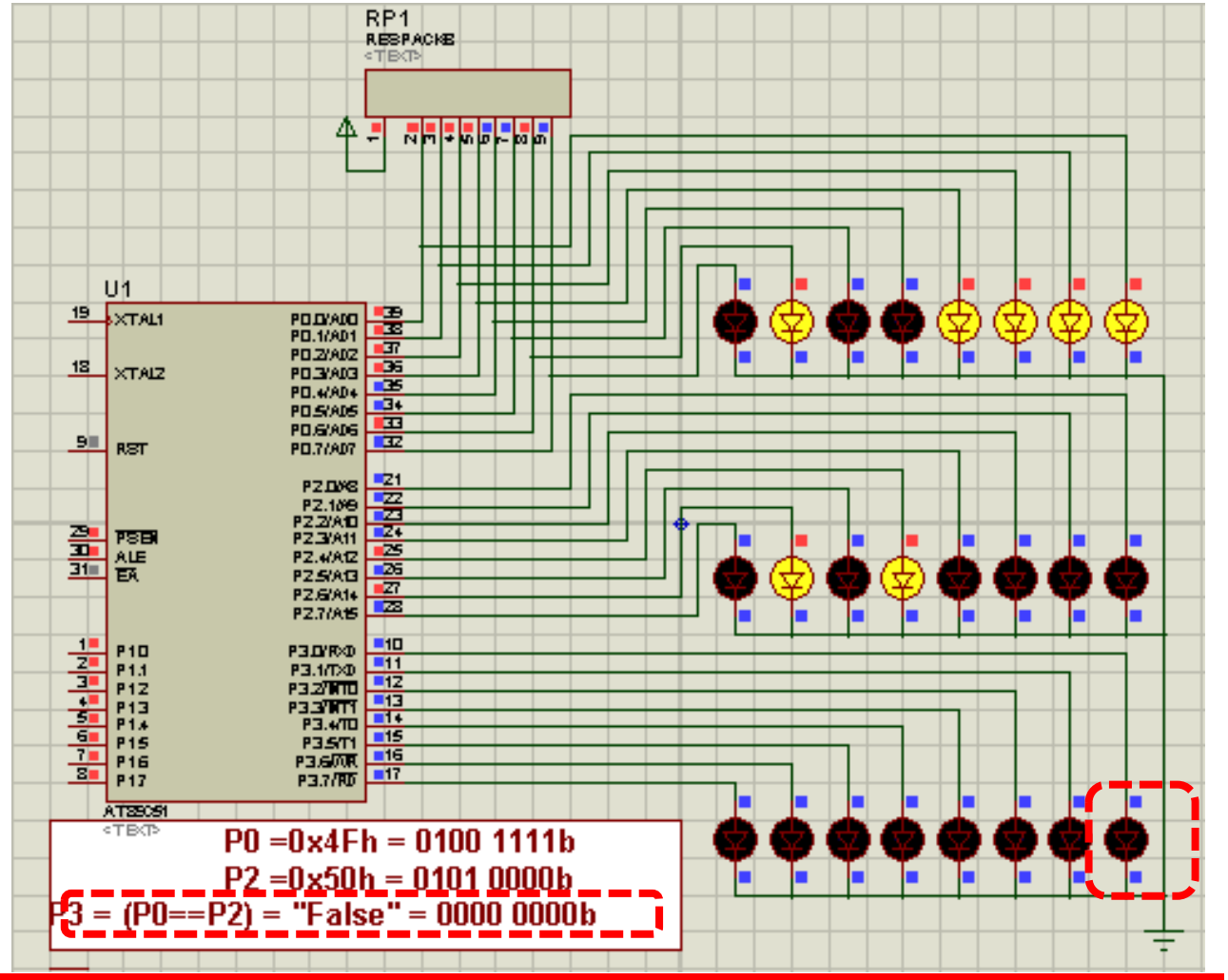


>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ
==	เท่ากับ
!=	ไม่เท่ากับ

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับการเปรียบเทียบ: > >= < <= == !=

```
#include<reg51.h>
void main()
{  P0=0x4F;
   P2=0x50;
   P3=(P0==P2); }
```

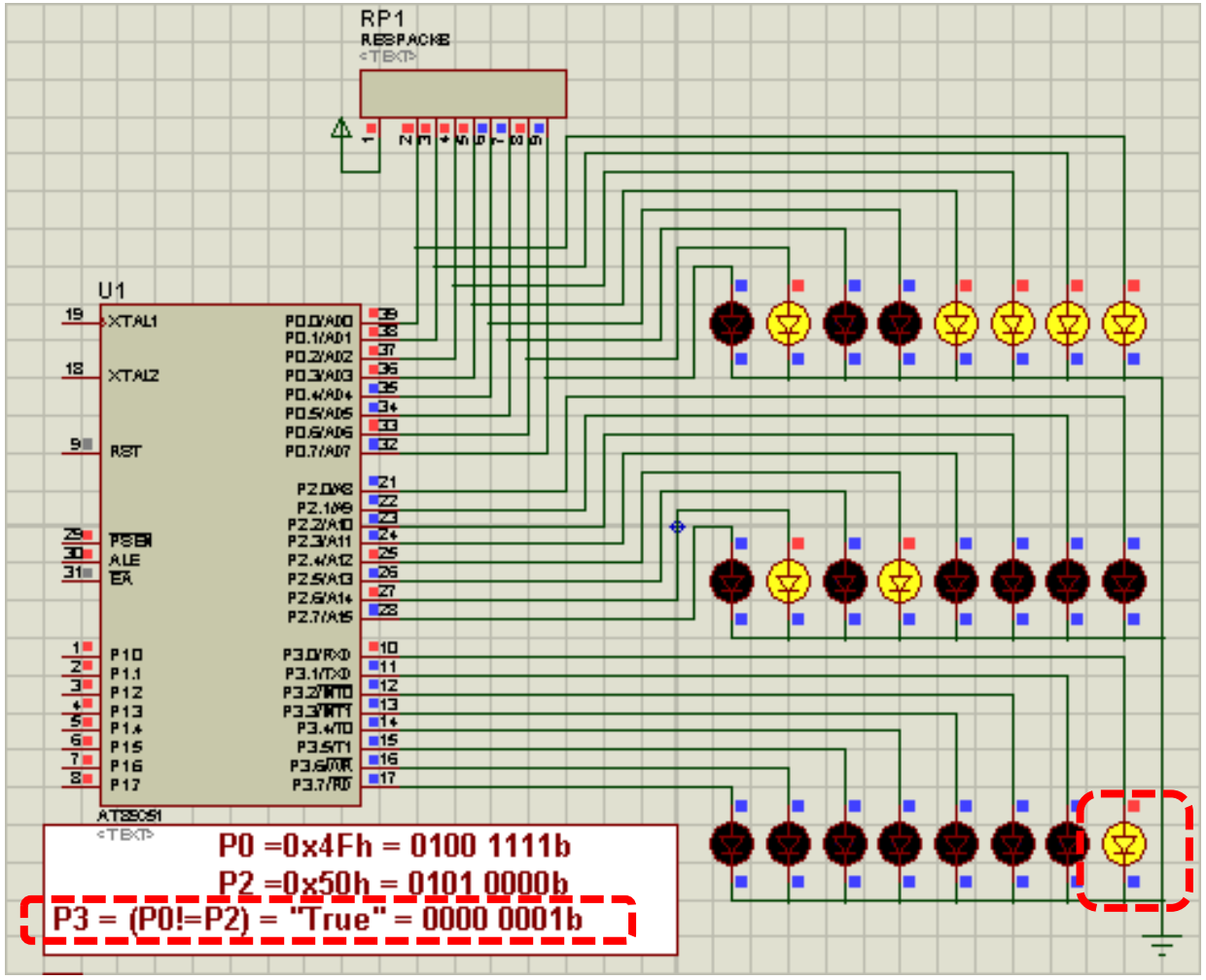


>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ
==	**เท่ากับ**
!=	ไม่เท่ากับ

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับการเปรียบเทียบ: > >= < <= == !=

```
#include<reg51.h>
void main()
{ P0=0x4F;
  P2=0x50;
  P3=(P0!=P2); }
```



>	มากกว่า
>=	มากกว่า หรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่า หรือเท่ากับ
==	เท่ากับ
!=	ไม่เท่ากับ

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

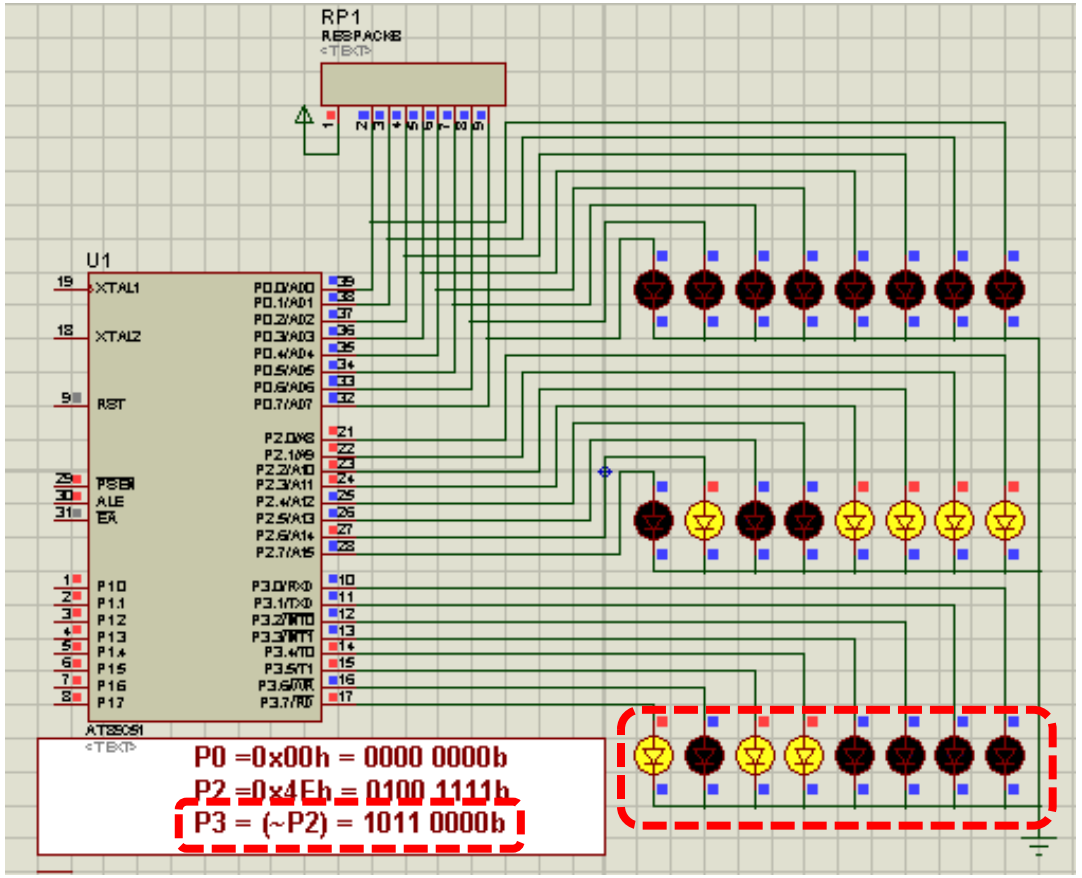
ตัวอย่างเกี่ยวกับการเปรียบเทียบ: ~ & | ^

Byte operation

- ~ นิเสธ(NOT operation)
- & และ (AND operation)
- | หรือ (OR operation)
- ^ (XOR operation)

```
#include<reg51.h>
void main()
{  P0=0x00;
   P2=0x4F;
   P3=(~P2); }
```

P2 = 0100 1111
P3 = 1011 0000



3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

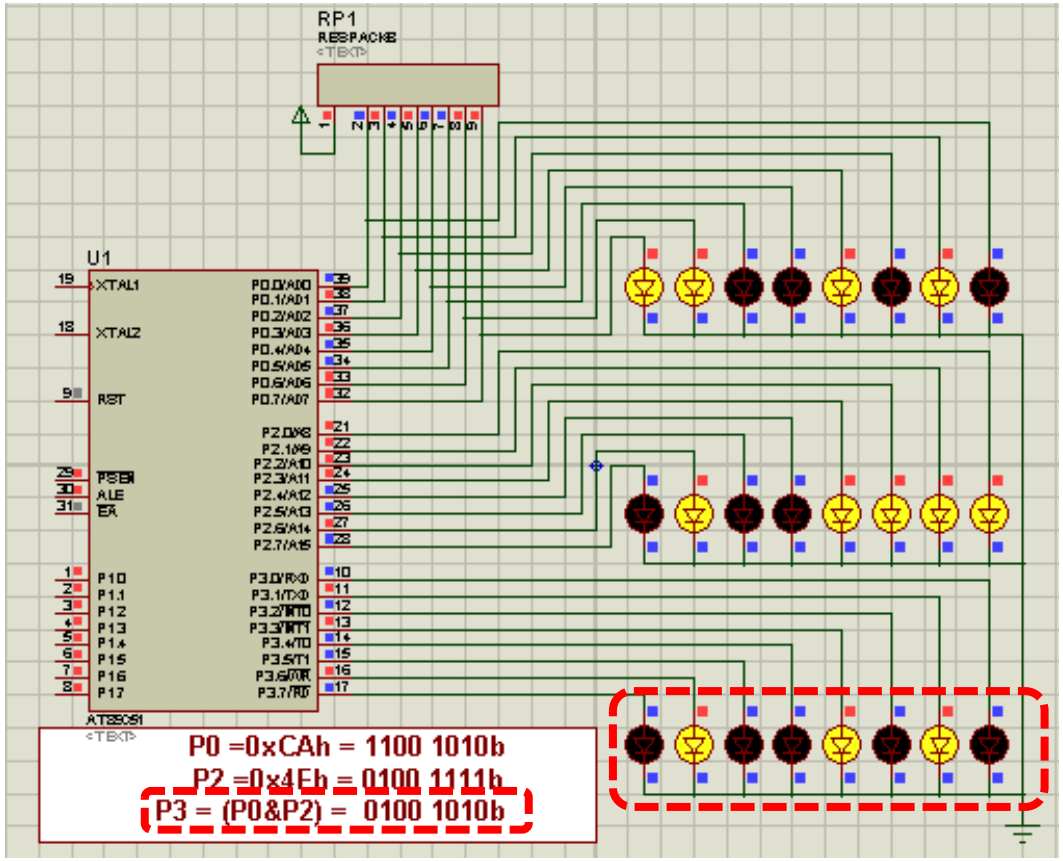
ตัวอย่างเกี่ยวกับตรรกศาสตร์: ~ & | ^

Byte operation

- ~ นิเสธ(NOT operation)
- & และ(AND operation)
- | หรือ (OR operation)
- ^ (XOR operation)

```
1  #include<reg51.h>
2  void main()
3  { P0=0xCA;
4    P2=0x4F;
5    P3=(P0&P2); }
6
```

P0 = 1100 1010
P2 = 0100 1111 &
P3 = 0100 1010



3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

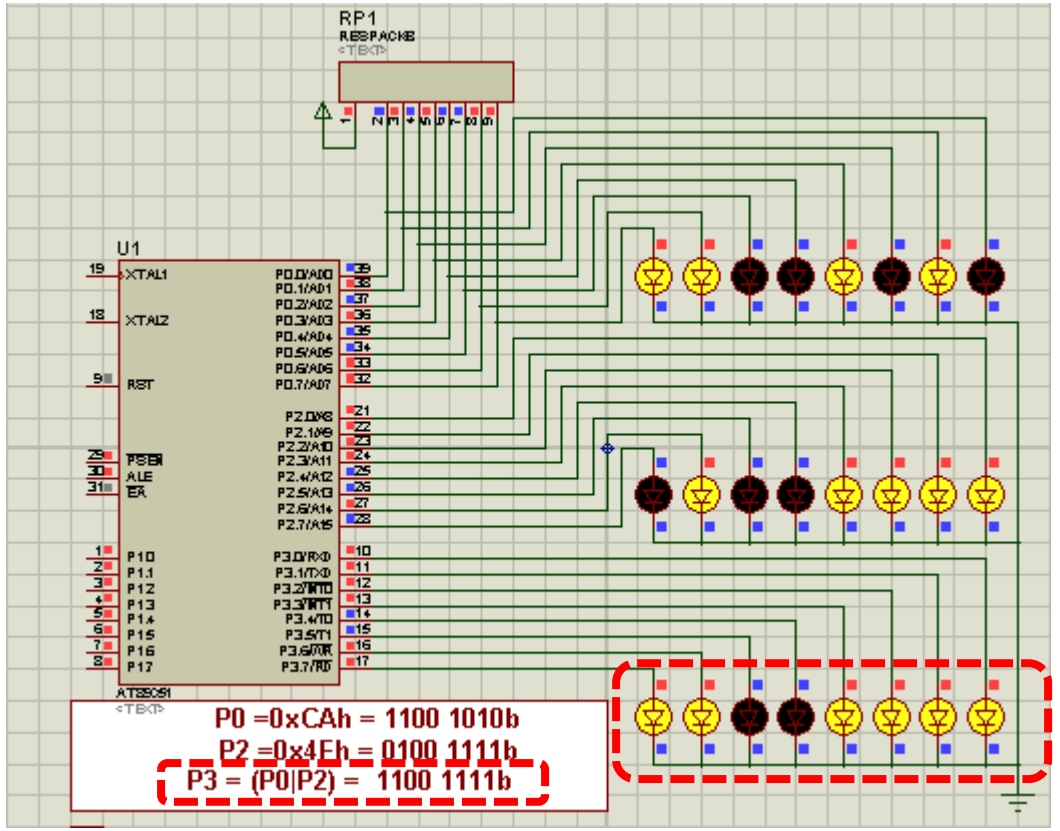
ตัวอย่างเกี่ยวกับตรรกศาสตร์: ~ & | ^

Byte operation

~	นิเสธ(NOT)
&	และ(AND operation)
|	**หรือ (OR operation)**
^	(XOR operation)

```
#include<reg51.h>
void main()
{  P0=0xCA;
   P2=0x4F;
   P3=(P0|P2); }
```

P0 = 1100 1010
P2 = 0100 1111
P3 = 1100 1111



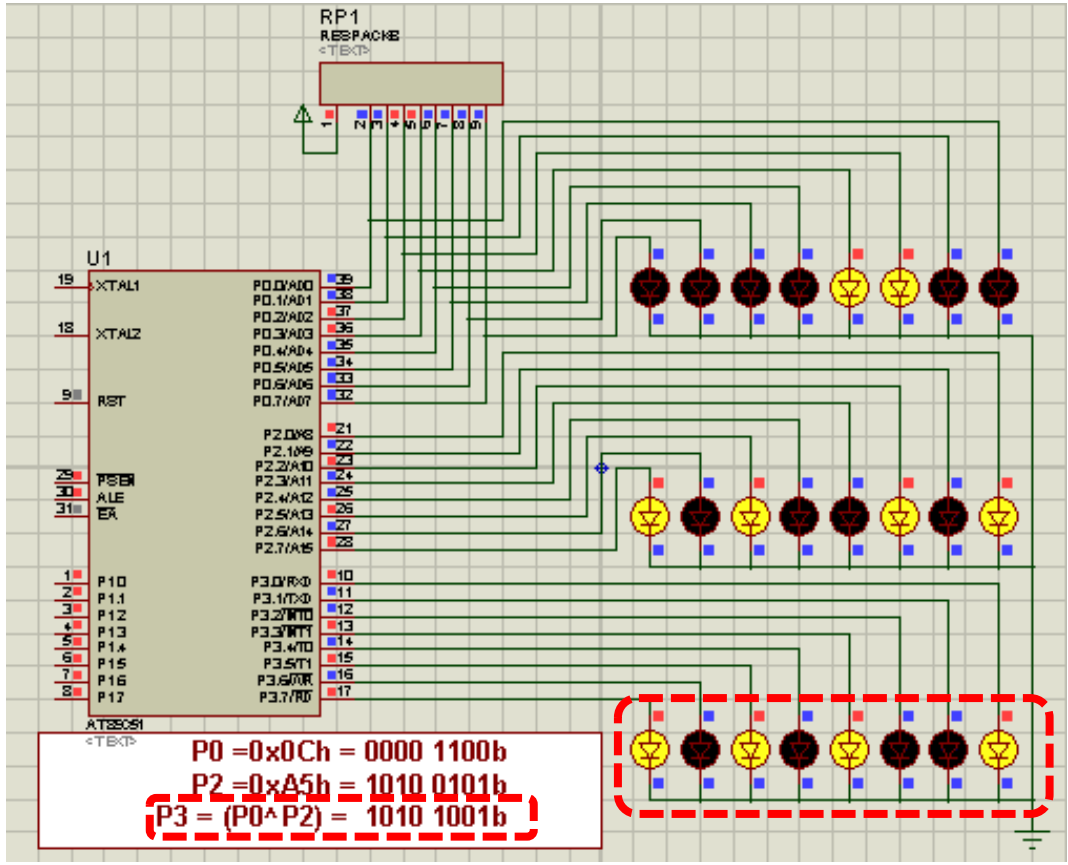
3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับตรรกศาสตร์: ~ & | ^

Byte operation	
~	นิเสธ(NOT)
&	และ(AND operation)
|	หรือ (OR operation)
^	(XOR operation)

```
1  #include<reg51.h>
2  void main()
3  { P0=0x0C;
4    P2=0xA5;
5    P3=(P0^P2); }
6
```

P0 = 0000 1100
P2 = 1010 0101 ^
P3 = 1010 1001



3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

ตัวอย่างเกี่ยวกับเลื่อนบิต >> <<

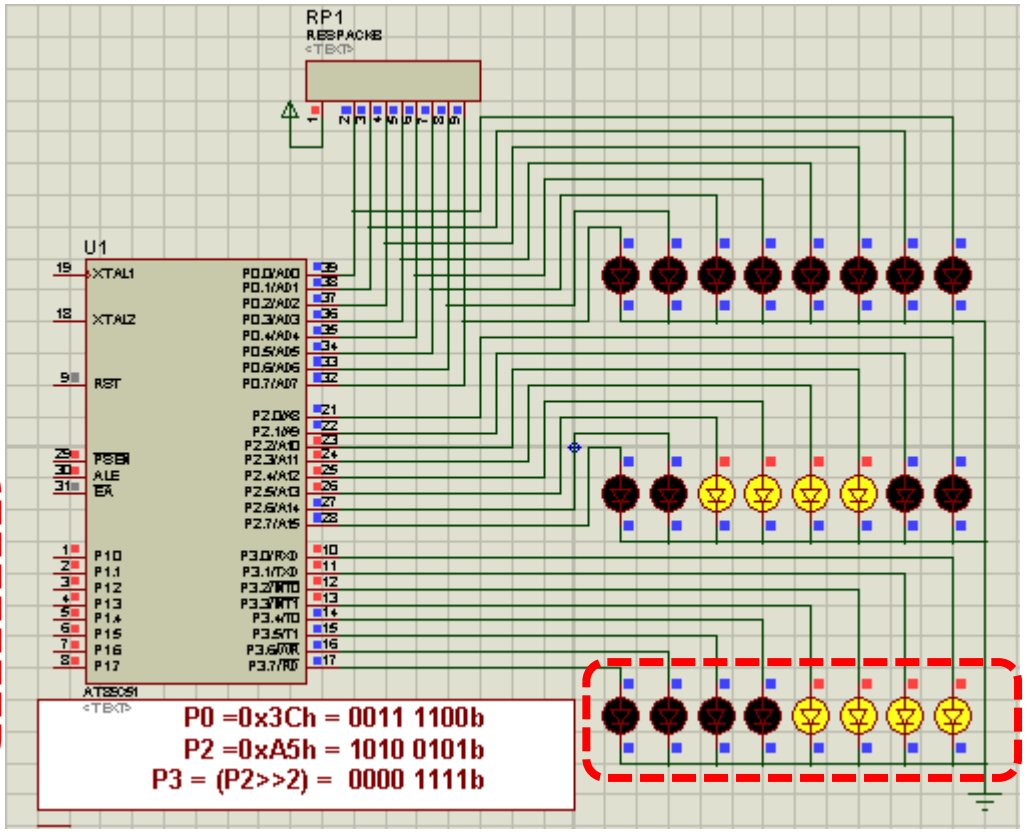
```
1  #include<reg51.h>
2  void main()
3  { P0=0x00;
4    P2=0x3C;
5    P3=(P2>>2); }
6
```

P2 = 0011 1100

P2>>1 = 0001 1110

P2>>2 = 0000 1111

P3 = 0000 1111



>> เลื่อนบิตไปทางขวา

<< เลื่อนบิตไปทางซ้าย

3.5 ชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์

ตัวอย่างเกี่ยวกับเลื่อนบิต >> <<

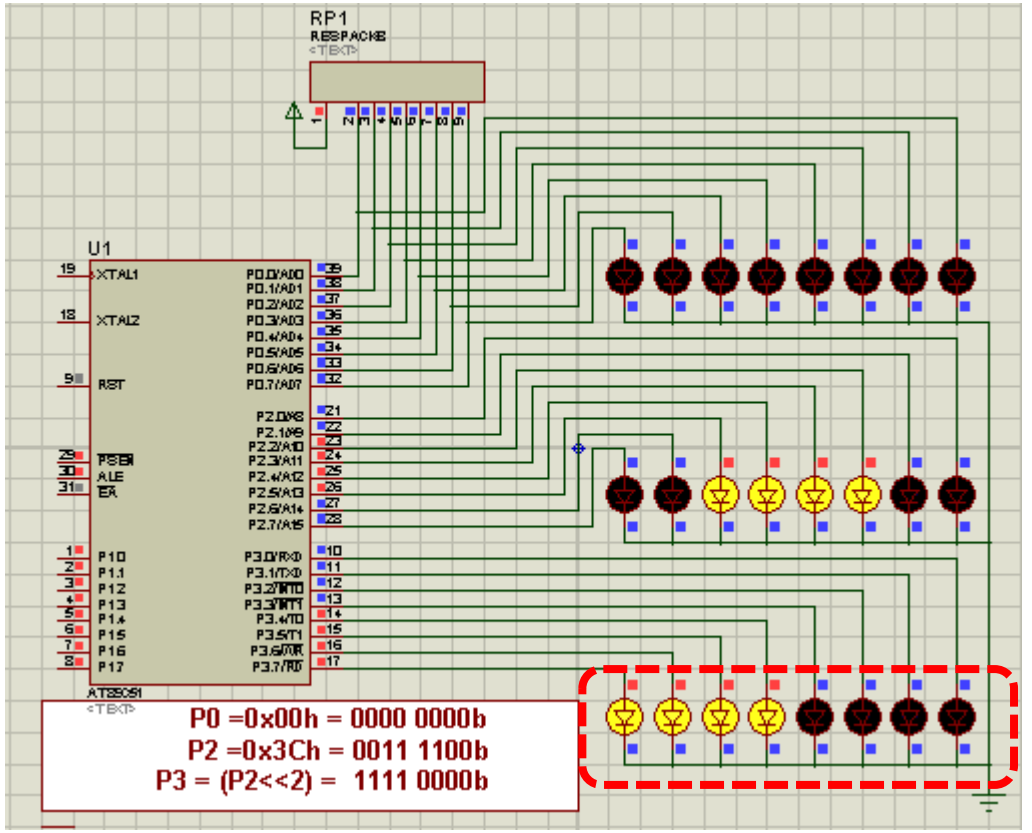
```
#include<reg51.h>
void main()
{ P0=0x00;
  P2=0x3C;
  P3=(P2<<2); }
```

P2 = 0011 1100

P2<<1 = 0111 1000

P2<<2 = 1111 0000

P3 = 1111 0000



>> เลื่อนบิตไปทางขวา

<< เลื่อนบิตไปทางซ้าย

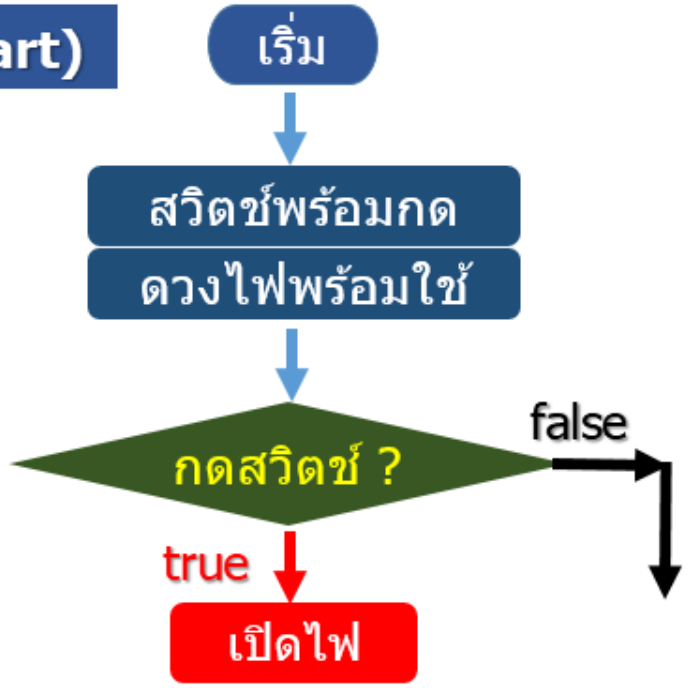
3.6(ก) ชุดคำสั่งเกี่ยวกับเงื่อนไข สองทางเลือก (if)

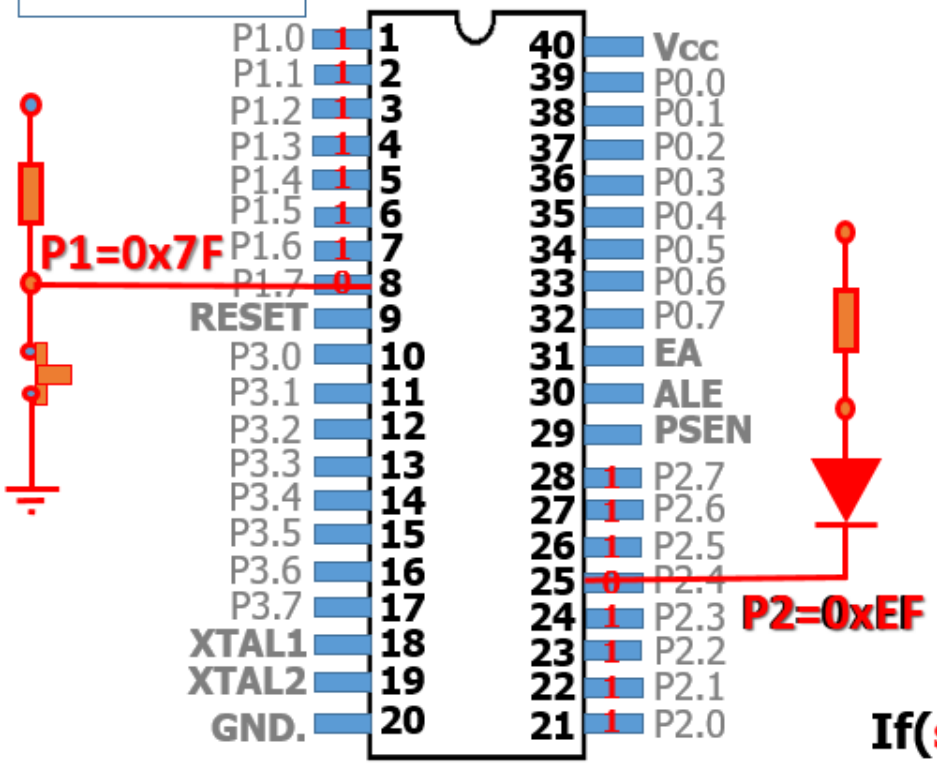
ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก สวิตช์เปิด/ปิด ดวงไฟ

วงจรสวิตช์ใช้ port P1.7

วงจรดวงไฟใช้ port P2.4

ผังความคิด (flowchart)






```
#include<reg51.h>
void main ()
{
  P1=0xFF;P2=0xFF;
  while(1)
    { if(P1==0x7F)
        {P2=0xEF;}


    }
}
```

รูปแบบโปรแกรม

If(statement is real) { do process }

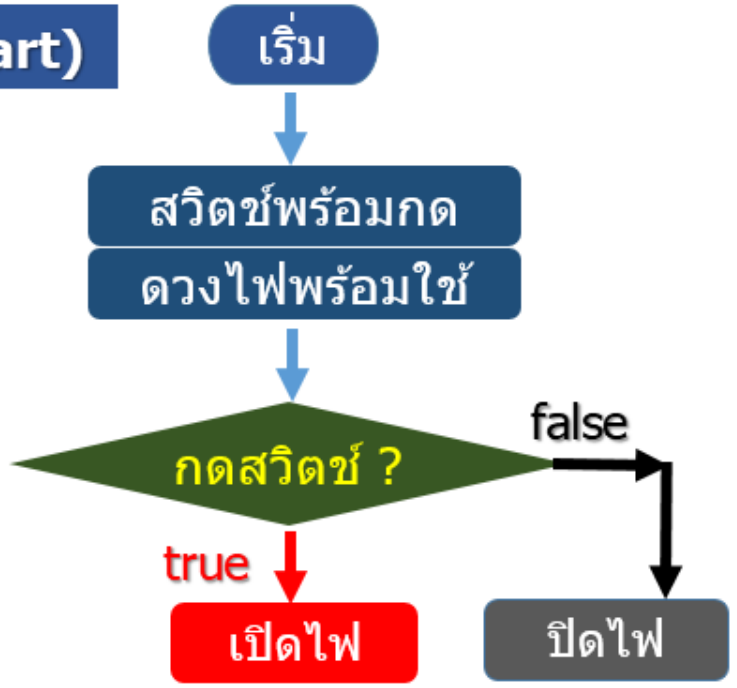
3.6(ก) ชุดคำสั่งเกี่ยวกับเงื่อนไข สองทางเลือก (if)

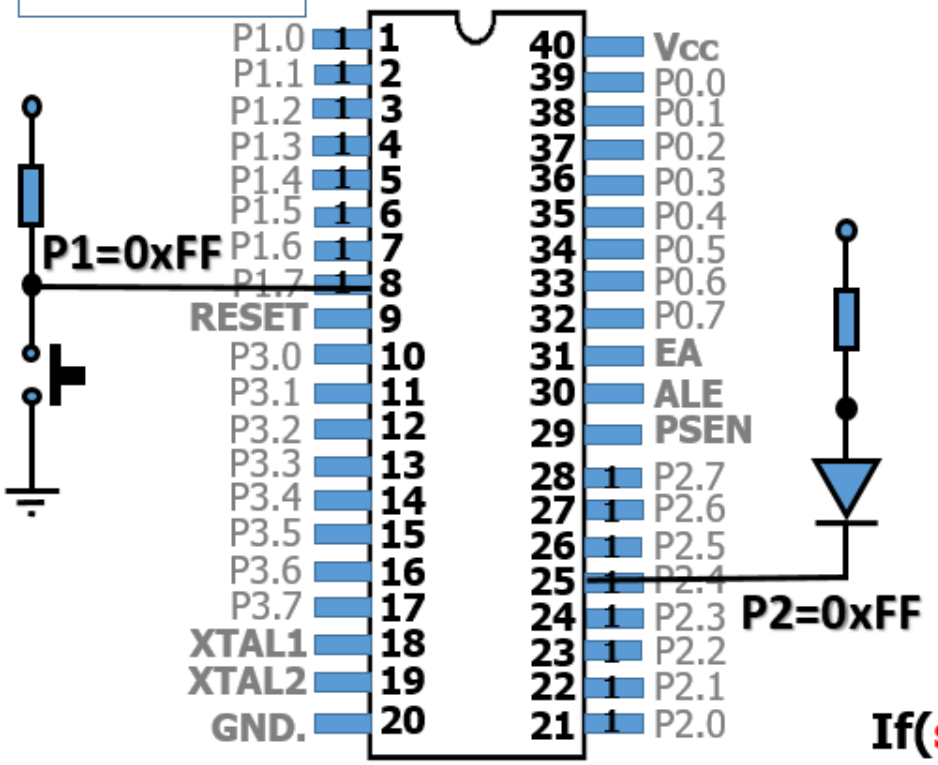
ตัวอย่างที่ใช้เงื่อนไข (if) สองทางเลือก สวิตช์เปิด/ปิด ดวงไฟ

วงจรสวิตช์ใช้ port P1.7

วงจรดวงไฟใช้ port P2.4

ผังความคิด (flowchart)






```
#include<reg51.h>
void main ()
{
  P1=0xFF;P2=0xFF;
  while(1)
     { if(P1==0x7F)
          {P2=0xEF;}
       else {P2=0xFF;}
     }
}
```

รูปแบบโปรแกรม

If(statement is real) { do process }

3.6(ข) ชุดคำสั่งเกี่ยวกับเงื่อนไข หลายทางเลือก (if)

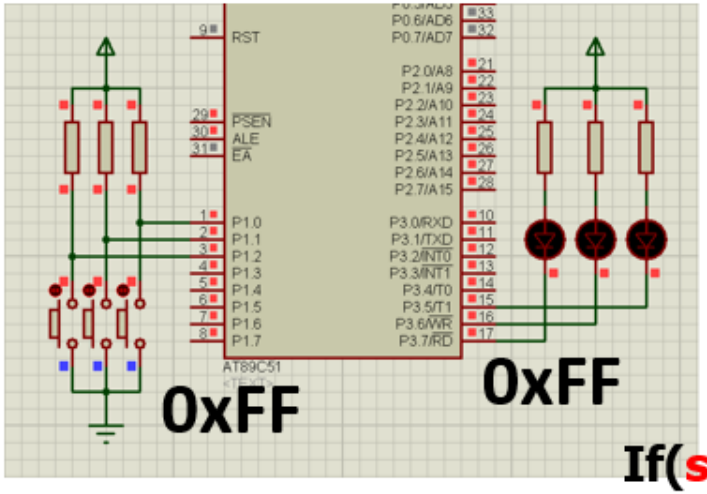
ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง

วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

ผังความคิด (flowchart)




```
#include<reg51.h>
 void main ()
{ Unsigned int xsw;
 while(1)
 { P1=0xFF;P2=0xFF;xsw=P1;
   }
}
```

รูปแบบโปรแกรม

If(statement is real) { do P1 } else{do P2}

3.6(ข) ชุดคำสั่งเกี่ยวกับเงื่อนไข หลายทางเลือก (if)

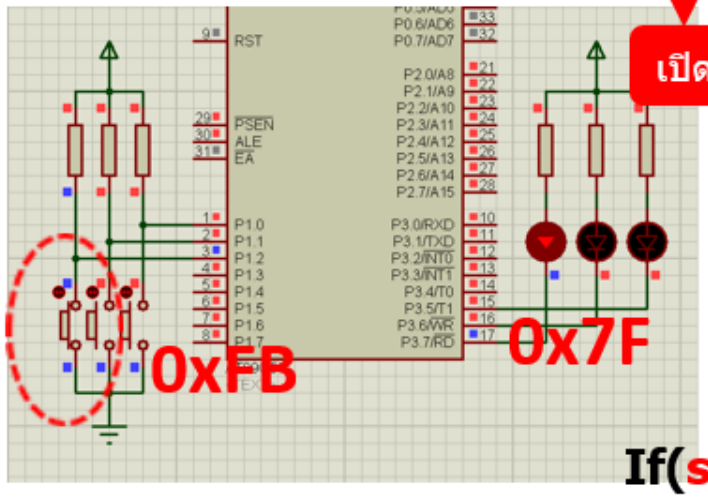
ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง

วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

ผังความคิด (flowchart)




```
#include<reg51.h>
 void main ()
{ Unsigned int xsw;
 while(1)
 { P1=0xFF;P2=0xFF;xsw=P1;
 if(xsw==0xFB){P3=0x7F;}
   }
}
```

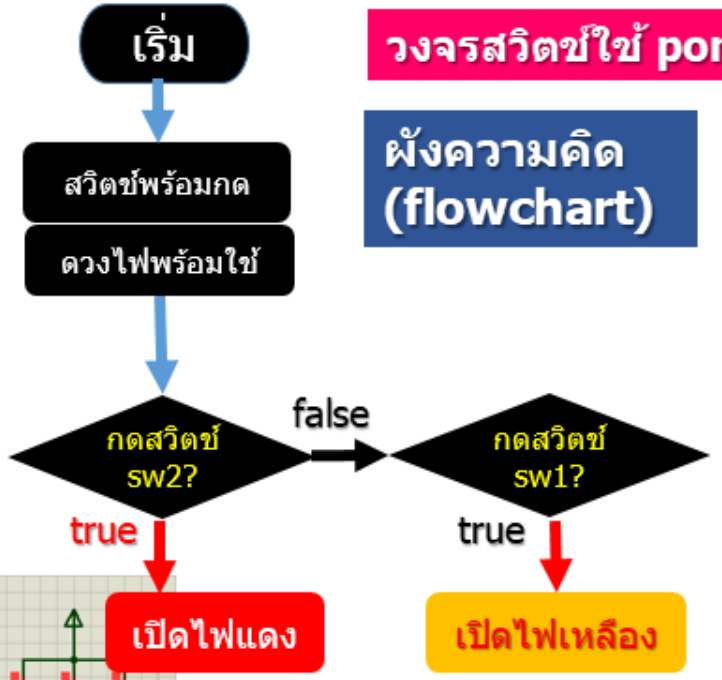
รูปแบบโปรแกรม

If(statement is real) { do P1 } else{do P2}

3.6(ข) ชุดคำสั่งเกี่ยวกับเงื่อนไข หลายทางเลือก (if)

ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

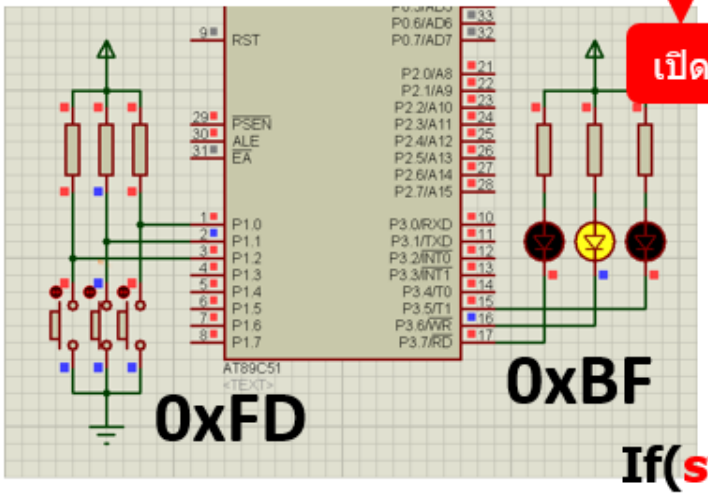
สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง



วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

ผังความคิด (flowchart)




```
#include<reg51.h>
 void main ()
{ Unsigned int xsw;
 while(1)
 { P1=0xFF;P2=0xFF;xsw=P1;
 if(xsw==0xFB){P3=0x7F;}
 else if(xsw==0xFD){P3=0xBF;}


   }
 }
```

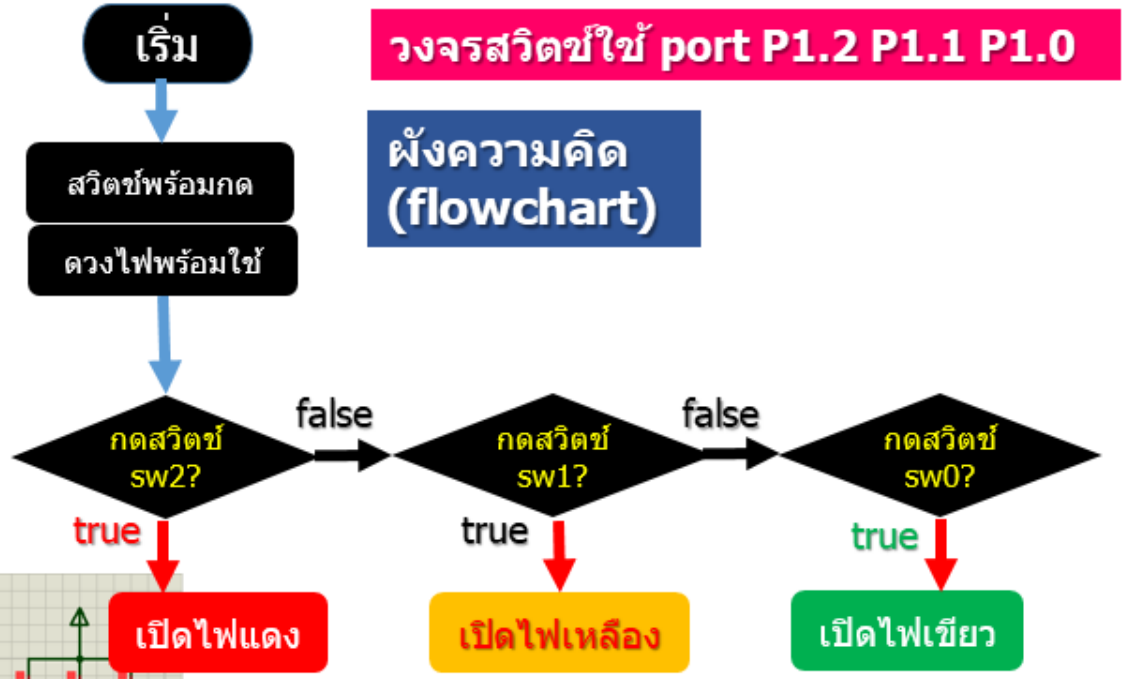
รูปแบบโปรแกรม

If(statement is real) { do P1 } else{do P2}

3.6(ข) ชุดคำสั่งเกี่ยวกับเงื่อนไข หลายทางเลือก (if)

ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

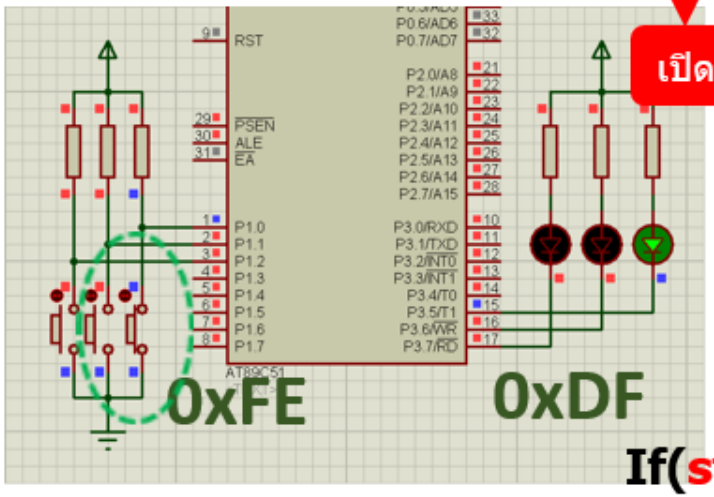
สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง



วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

ผังความคิด (flowchart)




```
#include<reg51.h>
 void main ()
{ Unsigned int xsw;
 while(1)
 { P1=0xFF;P2=0xFF;xsw=P1;
 if(xsw==0xFB){P3=0x7F;}
 else if(xsw==0xFD){P3=0xBF;}
 else if(xsw==0xFE){P3=0xDF;}
   }
}
```

รูปแบบโปรแกรม

If(statement is real) { do P1 } else{do P2}

3.6(ข) ชุดคำสั่งเกี่ยวกับเงื่อนไข หลายทางเลือก (if)

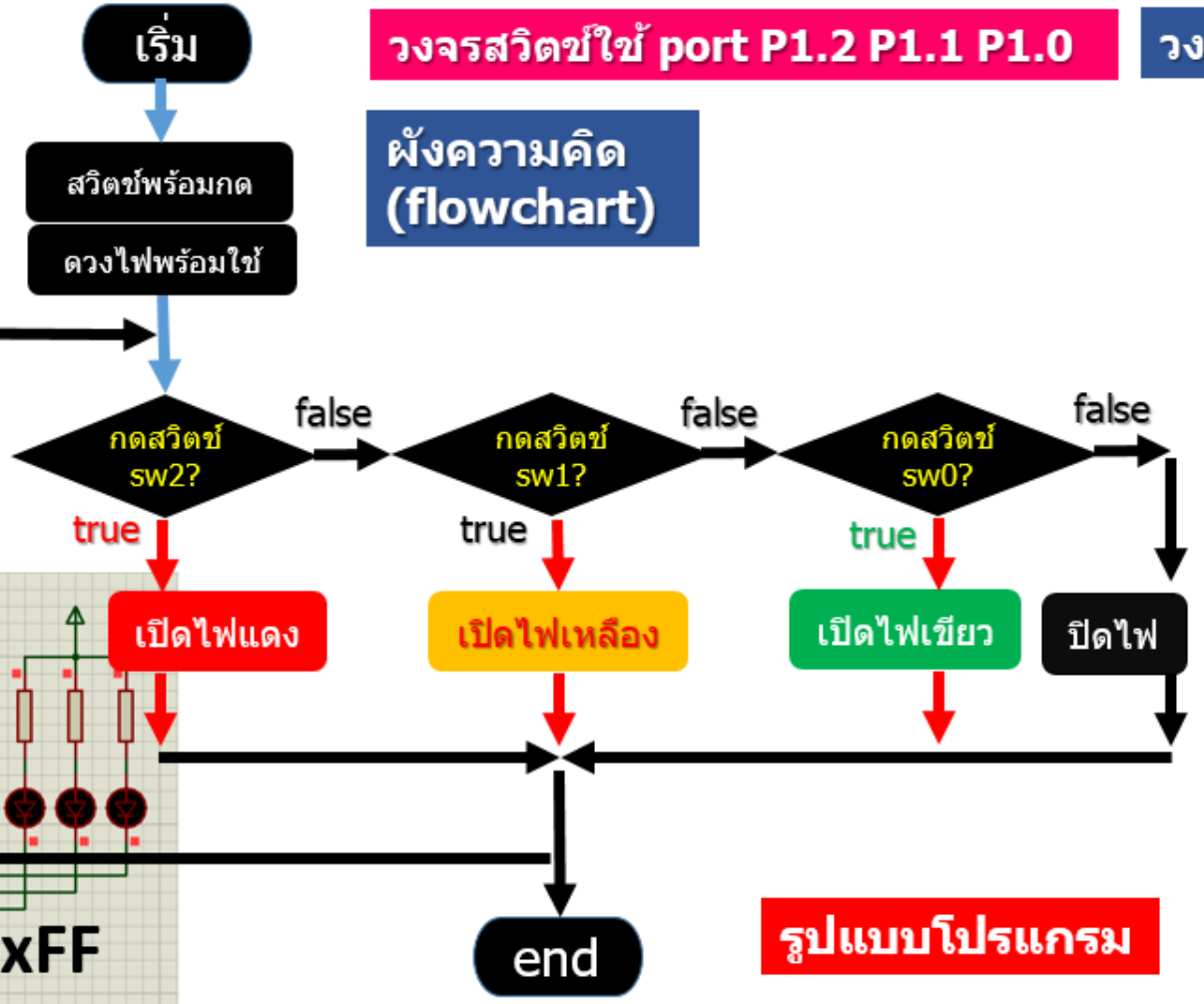
ตัวอย่างที่ใช้เงื่อนไข (if) หลายทางเลือก

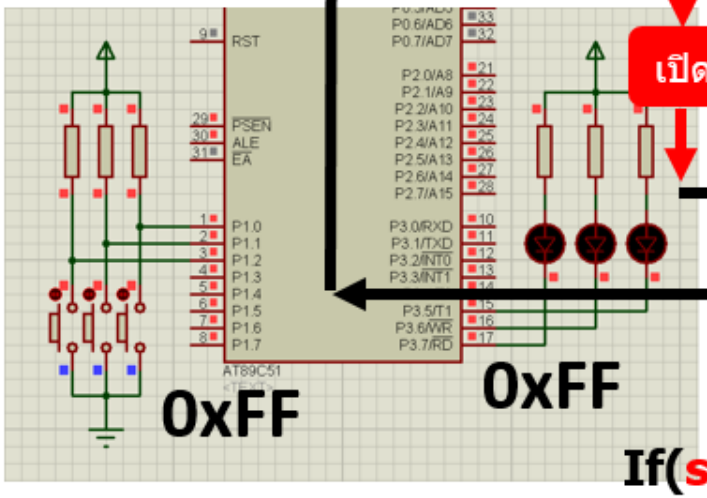
สวิตช์ 3 ตัวเปิด/ปิด ควบคุม ดวงไฟ 3 ดวง

วงจรสวิตช์ใช้ port P1.2 P1.1 P1.0

วงจรดวงไฟใช้ port P3.7 P3.6 P3.5

ผังความคิด (flowchart)






```
#include<reg51.h>
 void main ()
{ Unsigned int xsw;
 while(1)
 { P1=0xFF;P2=0xFF;xsw=P1;
 if(xsw==0xFB){P3=0x7F;}
 else if(xsw==0xFD){P3=0xBF;}
 else if(xsw==0xFE){P3=0xDF;}
 else {P3=0xFF;}
   }
 }
```

รูปแบบโปรแกรม

If(statement is real) { do P1 } else{do P2}

เมื่อจบบทเรียนนี้... นักศึกษาควรมีสมรรถนะดังนี้

3.1 เข้าใจโครงสร้างของโปรแกรมภาษา ซี

3.2 สามารถกำหนดและใช้งานชนิดตัวแปร และการตั้งชื่อตัวแปรในภาษาซี (Identifier)

3.3สามารถกำหนดและใช้งานตัวแปรชุด (Array Variable)

3.4 สามารถใช้งานชุดคำสั่งเกี่ยวกับการกำหนดค่า

3.5 สามารถใช้งานชุดคำสั่งเกี่ยวกับเลขคณิตและตรรกศาสตร์ & การเปรียบเทียบ

3.6 สามารถกำหนดและใช้งานชุดคำสั่งเกี่ยวกับเงื่อนไข (condition)

มหาวิทยาลัย
ราชภัฏนครปฐม