

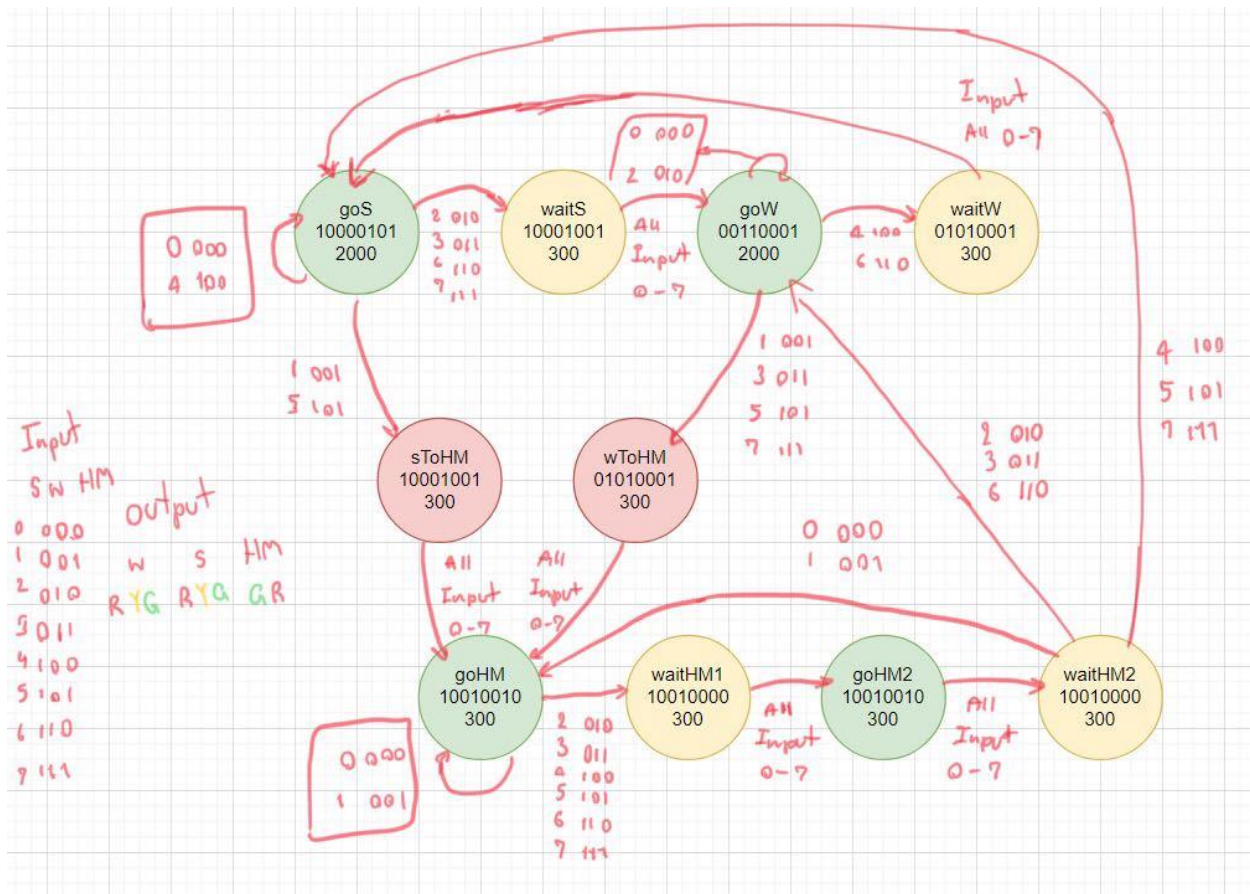
Assignment #5

ชื่อกลุ่ม : ฟ้ายรักพ่อ

สมาชิกในกลุ่ม : 64010270 นายคณพัฒน์ โพธิ์น้อย

64010315 นายธนธร แดงอ่อน

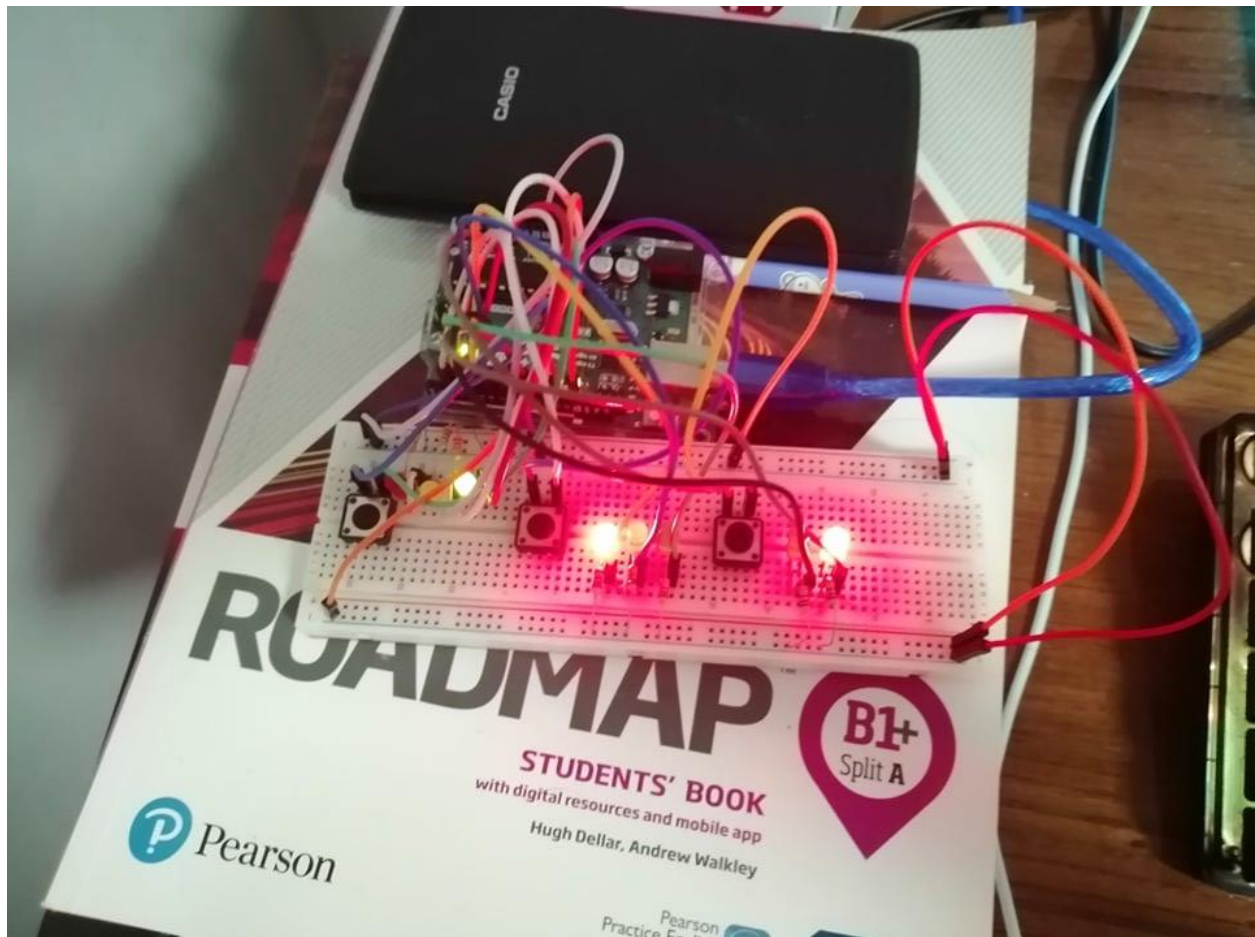
State Transition Graph



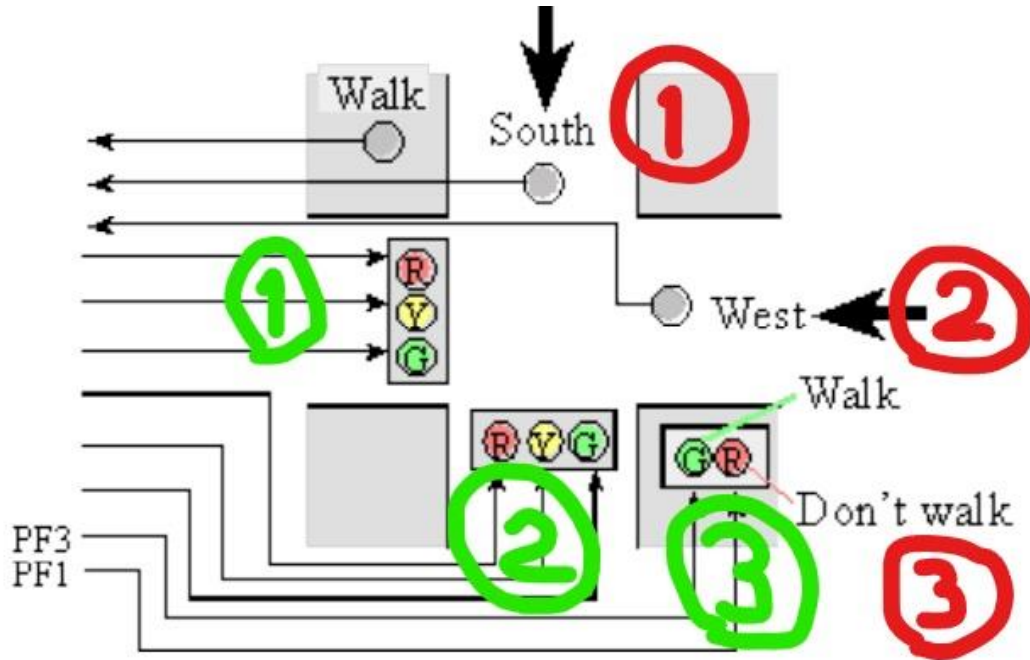
State Transition Table

Num	Name	Lights W S H M R Y G R Y G G R	Time	Input							
				0	1	2	3	4	5	6	7
				S W H M	S W H M	S W H M	S W H M	S W H M	S W H M	S W H M	S W H M
				0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	goS	100 001 01	2000	goS	sToHM	waitS	waitS	goS	sToHM	waitS	waitS
1	waitS	100 010 01	300	goW	goW	goW	goW	goW	goW	goW	goW
2	goW	001 100 01	2000	goW	wToHM	goW	wToK	waitW	wToHM	waitW	wToHM
3	waitW	010 100 01	300	goS	goS	goS	goS	goS	goS	goS	goS
4	sToHM	100 010 01	300	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1
5	wToHM	010 100 01	300	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1	goHM1
6	goHM1	100 100 10	300	goHM1	goHM1	waitHM1	waitHM1	waitHM1	waitHM1	waitHM1	waitHM1
7	waitHM1	100 100 00	300	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2	goHM2
8	goHM2	100 100 10	300	waitHM2	waitHM2	waitHM2	waitHM2	waitHM2	waitHM2	waitHM2	waitHM2
9	waitHM2	100 100 00	300	goHM1	goHM1	goW	goW	goS	goS	goW	goS

รูปถ่ายชิ้นงาน



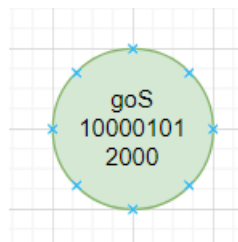
คำอธิบายโดยย่อ



- ตัว Input (South West Walk) ทำให้รู้ว่ามียอด หรือ คนรอข้ามถนนฝั่งไหน (1 = มี, 0 = ไม่มี) เช่น ถ้ามีรถมารอฝั่ง S แต่รถฝั่ง W ไม่มี และ ไม่มี HM ข้าม Input จึงเป็น 1 0 0 (S W HM)

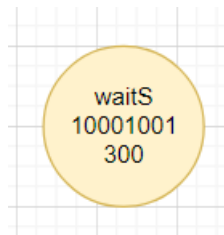
- ส่วนตัวของ Output (West South Walk) คือการแสดงผลของไฟจราจร บอกให้รถฝั่งใดหยุด หรือฝั่งใดไปได้ (ไฟดวงใดเปิด = 1 ไฟดวงใดปิด = 0) เช่น ถ้าตัว Input เป็น 1 0 0 (รถมารอฝั่ง S แต่รถฝั่ง W ไม่มี และ ไม่มี HM ข้าม) Output ที่จะแสดงผลคือ 100 001 01 (W S HM)

- เริ่มต้นที่ State 0 ทำงานเป็นอันดับ 1 (ถ้าไม่มีการกดปุ่มเกิดขึ้นมันก็จะวนอยู่ที่ State 0)



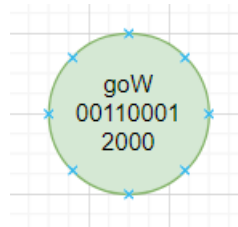
State 0 = goS (S = 100, W = 001, HM = 01)

- ถ้า Input เพิ่ม เป็น 0 1 0 หรือ 0 1 1 หรือ 1 1 0 หรือ 1 1 1 อันใดอันหนึ่ง



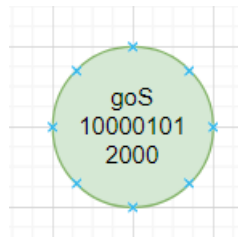
จาก State 0 -> State 1 คือ waitS (S = 100, W = 010, HM = 01) ได้ Output = (W = 100, S = 010, HM = 01)

-หลังจากนั้น State 1 จะกลายเป็น State 2 อัตโนมัติ ถ้าจะใส่ Input ใหนเข้ามา State ก็ยังไม่เปลี่ยนแปลง เป็น State 3

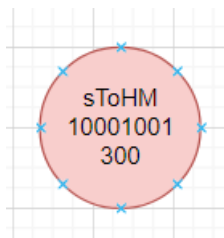


State 2 จะ Output = (W = 001, S = 100, HM = 01)

- การทำงานของ State 0 -> State 1 -> State 2 จะคล้ายๆการทำงานตั้งแต่ State 0 -> State 3 จะทำงาน เป็น Loop เหมือนกัน แต่ถ้ากด Walk Button เช่น

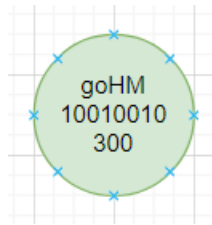


เริ่มต้นที่ State 0 เมื่อกด Input = 001 หรือ 101 อันใดอันหนึ่ง ทำให้ State 0 -> State 4

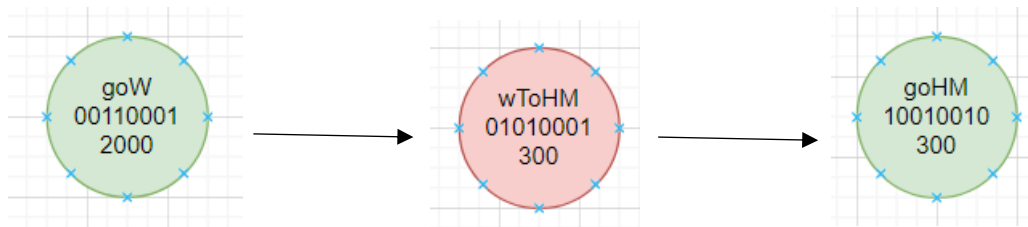


Output = (W = 100, S = 010, HM = 01) ให้รถฝั่ง South เตรียมหยุดรถ จึงให้เปิดไฟเหลือง

- หลังจากนั้น State 4 -> State 6 อัตโนมัต ถ้าใส่ Input ไหน มันก็ทำงานที่ State 6 อยู่ ก็จะได้ Output =
(W = 100, S = 100, HM = 10)



- การทำงานนี้จะคล้ายกับการทำงาน State 2 -> State 5 -> State 6

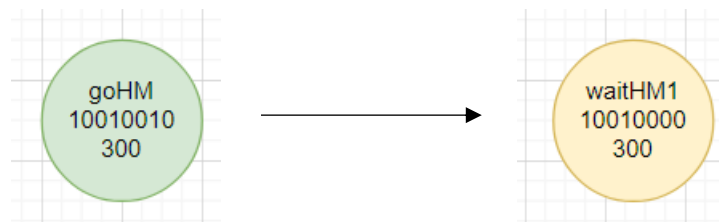


- ถ้าใส่ Input = 0 0 1 หรือ 0 1 1 หรือ 1 0 1 หรือ 1 1 1

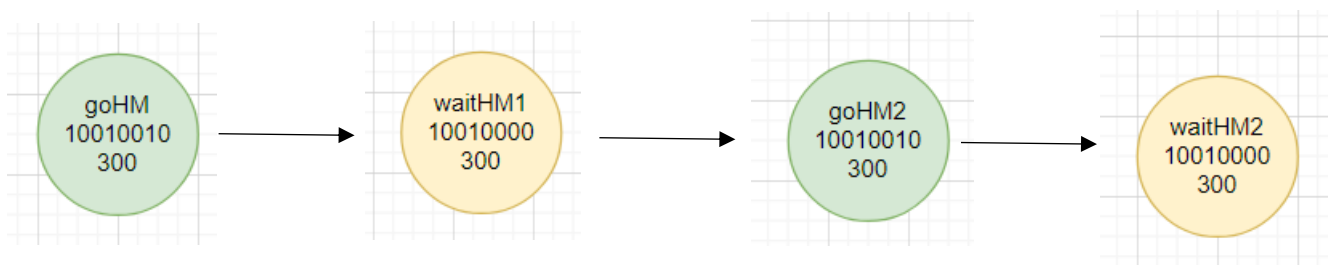
- State 2 -> State 5

- State 5 -> State 6 อัตโนมัต ถ้าใส่ Input ไหน มันก็ทำงานที่ State 6 อยู่

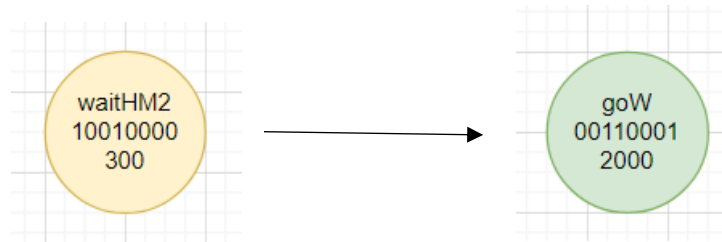
- หลังจาก State 6 ถ้ากด Button แล้วมีรถฝั่ง West จะ Input 0 1 0 หรือ 0 1 1 หรือ 1 0 0 หรือ 1 0 1 หรือ 1 1 1 อันใดอันหนึ่ง ทำให้ State 6 -> State 7



Output = (W = 100, S = 100, HM = 00) เพื่อให้ไฟเขียวทำงานเป็นไฟกระพริบ แบบ 0 1 0 (State7, State8, State9) และทำงาน -> State 2 หรือ State 0

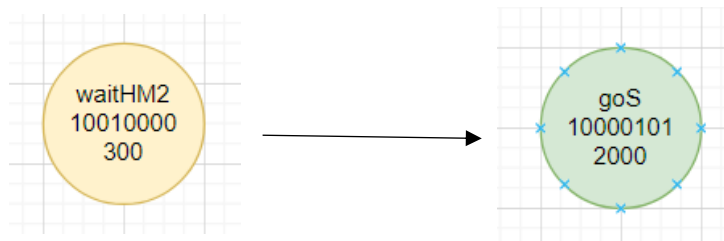


- Input 6 ตัวนี้ 1 ใน Input ถูกเลือก 0 1 0 หรือ 0 1 1 หรือ 1 1 0 อันใดอันหนึ่ง ส่ง State 9 -> State 2



Output = (W = 001, S = 100, HM = 01)

- Input 6 ตัวนี้ 1 ใน Input ถูกเลือก 1 0 0 หรือ 1 0 1 หรือ 1 1 1 อันใดอันหนึ่ง ส่ง State 9 -> State 0



Output = (W = 100, S = 001, HM = 01)

```

#define W_R_LED 5
#define W_Y_LED 6
#define W_G_LED 7
#define WEST_Button 11

#define S_R_LED 2
#define S_Y_LED 3
#define S_G_LED 4
#define SOUTH_Button 10

#define HM_G_LED 8
#define HM_R_LED 9
#define WALK_Button 12

#define goS      0
#define waitS    1
#define goW      2
#define waitW    3
#define sToHM    4
#define wToHM    5
#define goHM1    6
#define waitHM1  7
#define goHM2    8
#define waitHM2  9

struct State
{
    unsigned long ST_Out;    // 6-bit pattern to street
output
    unsigned long Time;      // delay in milli seconds
units
    unsigned long Next[8];   // next state for inputs
0,1,2,3,4,5,6,7

```

```
};
```

```
typedef const struct State SType;
```

```
SType FSM[10] =
```

```
{
```

```
    {B10000101, 2000, {0, 4, 1, 1, 0, 4, 1, 1}}, //
```

```
State0 = {goS, SToHM, waitS, waitS, goS, sToHM, waitS,  
waitS}
```

```
    {B10001001, 300, {2, 2, 2, 2, 2, 2, 2, 2}}, //
```

```
State1 = {goW, goW, goW, goW, goW, goW, goW, goW}
```

```
    {B00110001, 2000, {2, 5, 2, 5, 3, 5, 3, 5}}, //
```

```
State2 = {goW, wToHM, goW, wToHM, waitW, wToHM, waitW,  
wToHM}
```

```
    {B01010001, 300, {0, 0, 0, 0, 0, 0, 0, 0}}, //
```

```
State3 = {goS, goS, goS, goS, goS, goS, goS, goS}
```

```
    {B10001001, 300, {6, 6, 6, 6, 6, 6, 6, 6}}, //
```

```
State4 = {goHM1, goHM1, goHM1, goHM1, goHM1, goHM1,  
goHM1, goHM1}
```

```
    {B01010001, 300, {6, 6, 6, 6, 6, 6, 6, 6}}, //
```

```
State5 = {goHM1, goHM1, goHM1, goHM1, goHM1, goHM1,  
goHM1, goHM1}
```

```
    {B10010010, 300, {6, 6, 7, 7, 7, 7, 7, 7}}, //
```

```
State6 = {goHM1, goHM1, waitHM1, waitHM1, waitHM1,  
waitHM1, waitHM1, waitHM1}
```

```
    {B10010000, 300, {8, 8, 8, 8, 8, 8, 8, 8}}, //
```

```
State7 = {goHM2, goHM2, goHM2, goHM2, goHM2, goHM2,  
goHM2, goHM2}
```

```
    {B10010010, 300, {9, 9, 9, 9, 9, 9, 9, 9}}, //
```

```
State8 = {waitHM2, waitHM2, waitHM2, waitHM2, waitHM2,  
waitHM2, waitHM2, waitHM2}
```

```
    {B10010000, 300, {6, 6, 2, 2, 0, 0, 2, 0}} //
```

```
State9 = {goHM1, goHM1, goW, goW, goS, goS, goW, goS}
```



```
};
```

```
unsigned long S = 0;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
    pinMode(W_R_LED, OUTPUT);
```

```
    pinMode(W_Y_LED, OUTPUT);
```

```
    pinMode(W_G_LED, OUTPUT);
```

```
    pinMode(WEST_Button, INPUT_PULLUP);
```

```
    pinMode(S_R_LED, OUTPUT);
```

```
    pinMode(S_Y_LED, OUTPUT);
```

```
    pinMode(S_G_LED, OUTPUT);
```

```
    pinMode(SOUTH_Button, INPUT_PULLUP);
```

```
    pinMode(HM_G_LED, OUTPUT);
```

```
    pinMode(HM_R_LED, OUTPUT);
```

```
    pinMode(WALK_Button, INPUT_PULLUP);
```

```
}
```

```
int input, input1, input2, input3;
```

```
void loop()
```

```
{
```

```
    digitalWrite(W_R_LED, (FSM[S].ST_Out & B10000000));
```

```
    digitalWrite(W_Y_LED, (FSM[S].ST_Out & B01000000));
```

```
    digitalWrite(W_G_LED, (FSM[S].ST_Out & B00100000));
```

```
    digitalWrite(S_R_LED, (FSM[S].ST_Out & B00010000));
```

```
    digitalWrite(S_Y_LED, (FSM[S].ST_Out & B00001000));
```

```
    digitalWrite(S_G_LED, (FSM[S].ST_Out & B00000100));
```

```
digitalWrite(HM_G_LED, (FSM[S].ST_Out & B00000010));
digitalWrite(HM_R_LED, (FSM[S].ST_Out & B00000001));

delay(FSM[S].Time);

input1 = !digitalRead(SOUTH_Button);
input2 = !digitalRead(WEST_Button);
input3 = !digitalRead(WALK_Button);

input = 4 * input2 + 2 * input1 + input3;
S = FSM[S].Next[input];
}
```