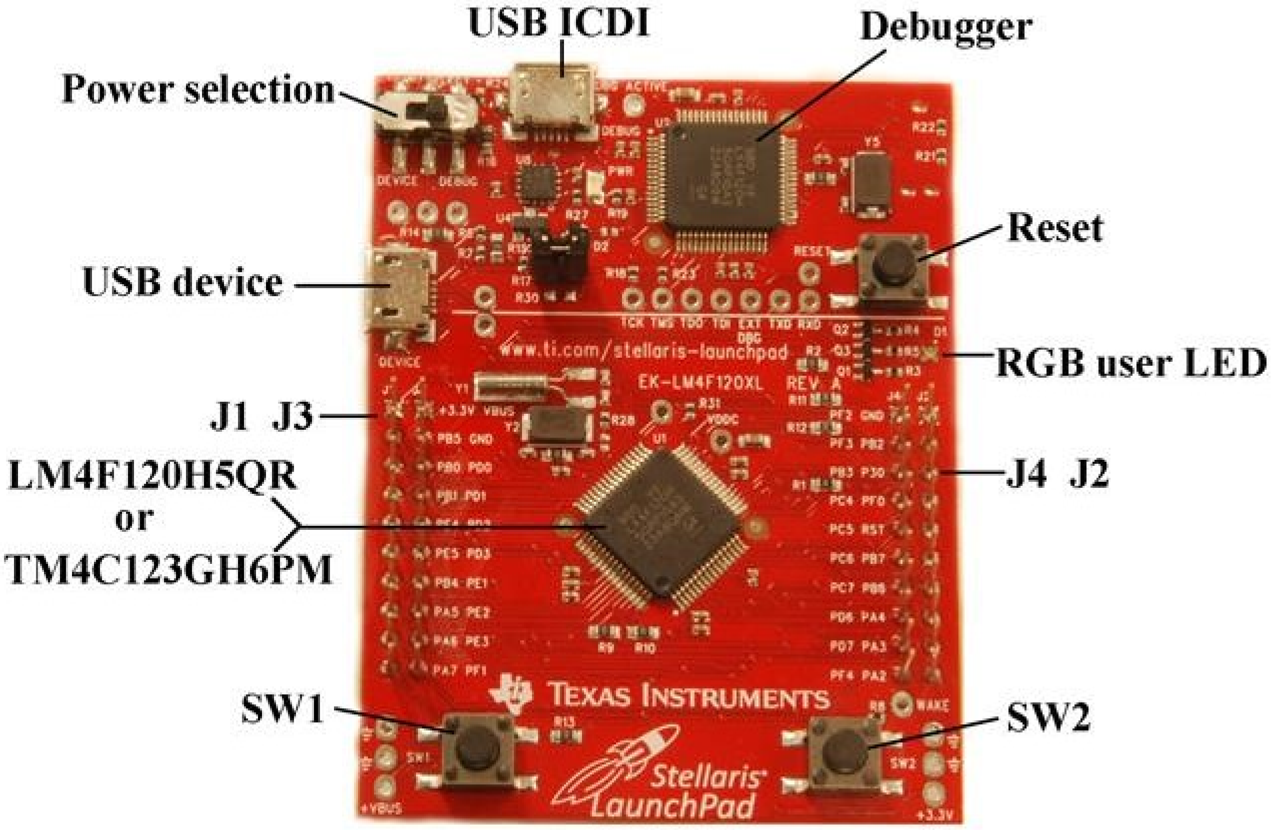
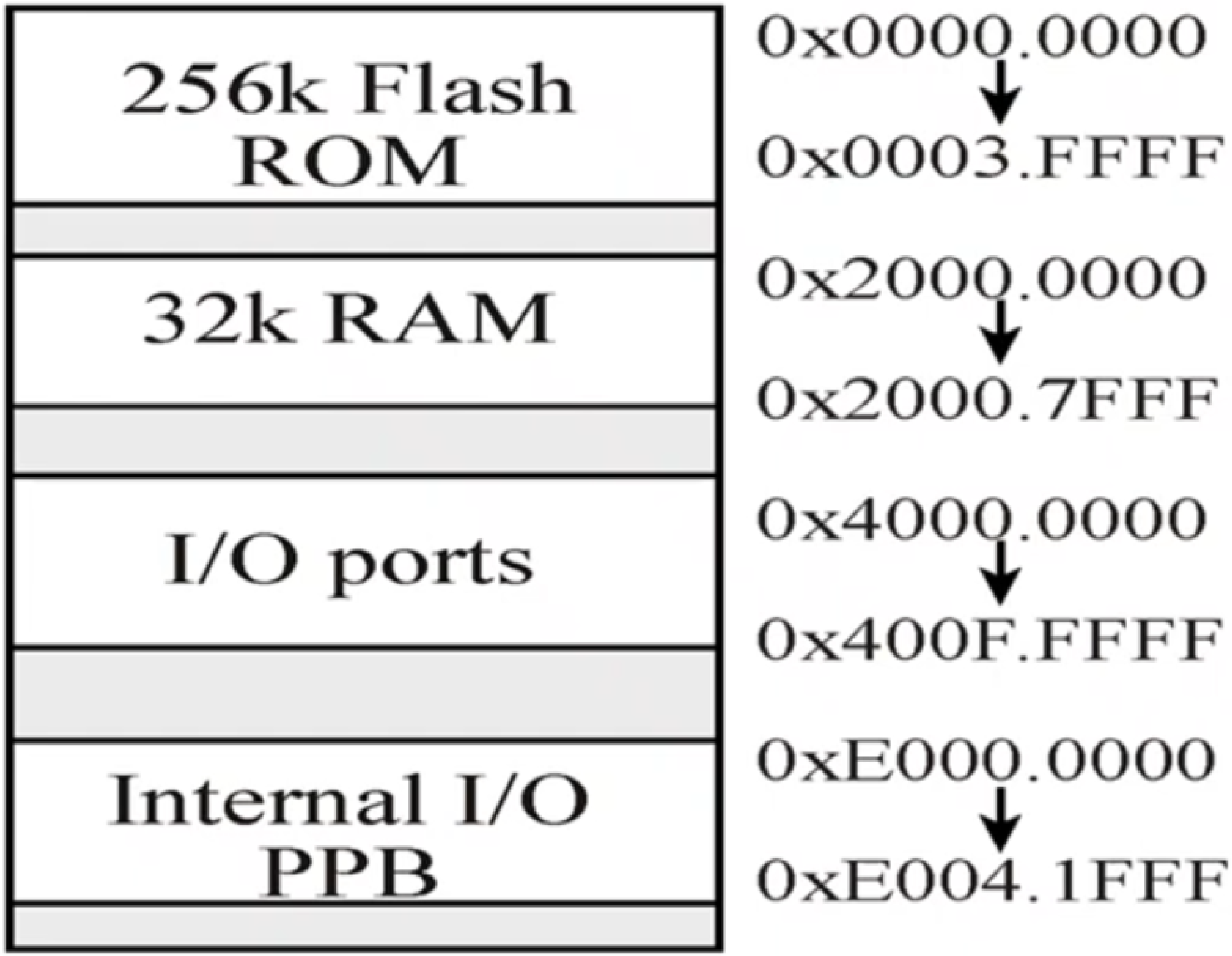
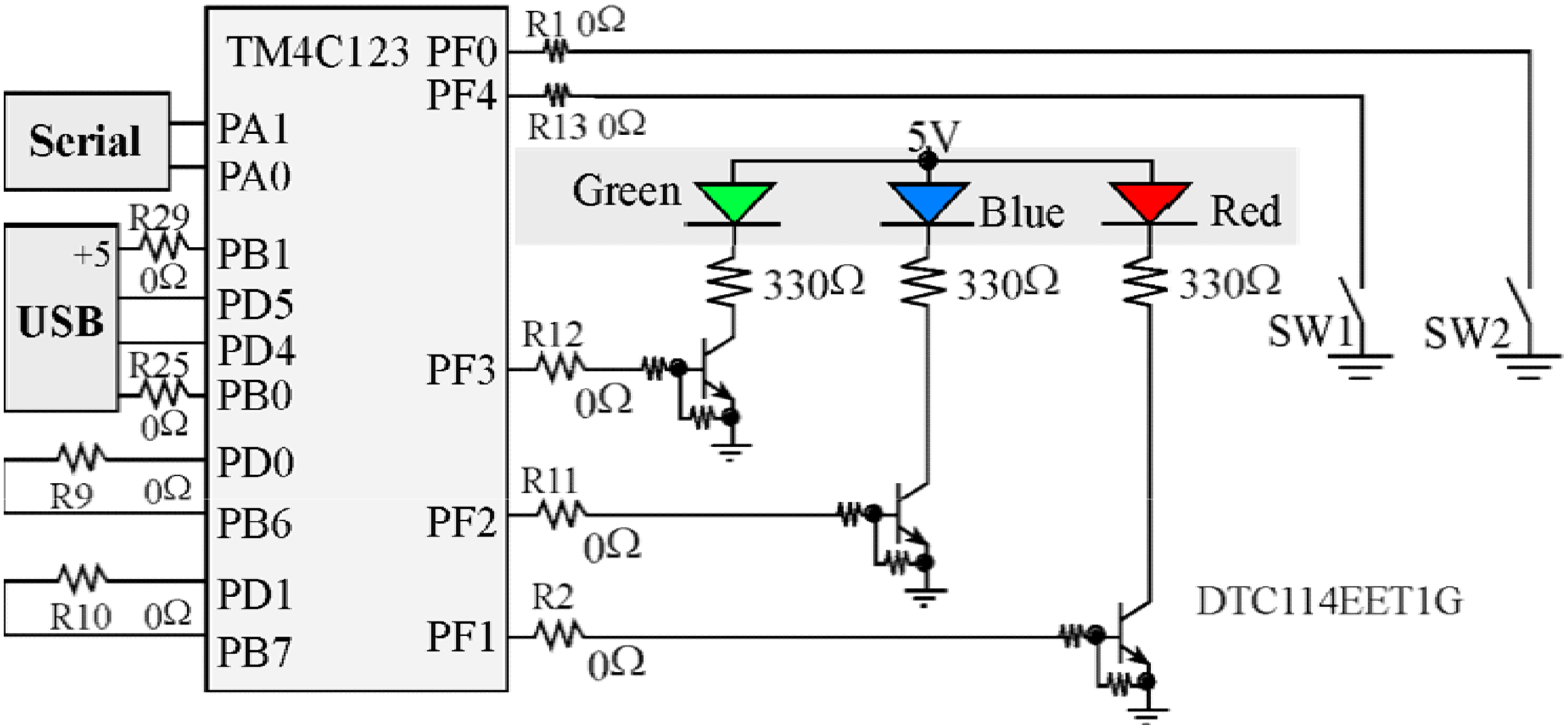
Tiva C Board


USB ICDI
Debugger
Power selection
Reset
USB device
RGB user LED
J1 J3
J4 J2
LM4F120H5QR
or
TM4C123GH6PM
SW1
SW2
www.ti.com/stellaris-launchpad
EK-LM4F120XL REV A
TEXAS INSTRUMENTS
Stellaris
LaunchPad

TI TM4C123 Memory-map

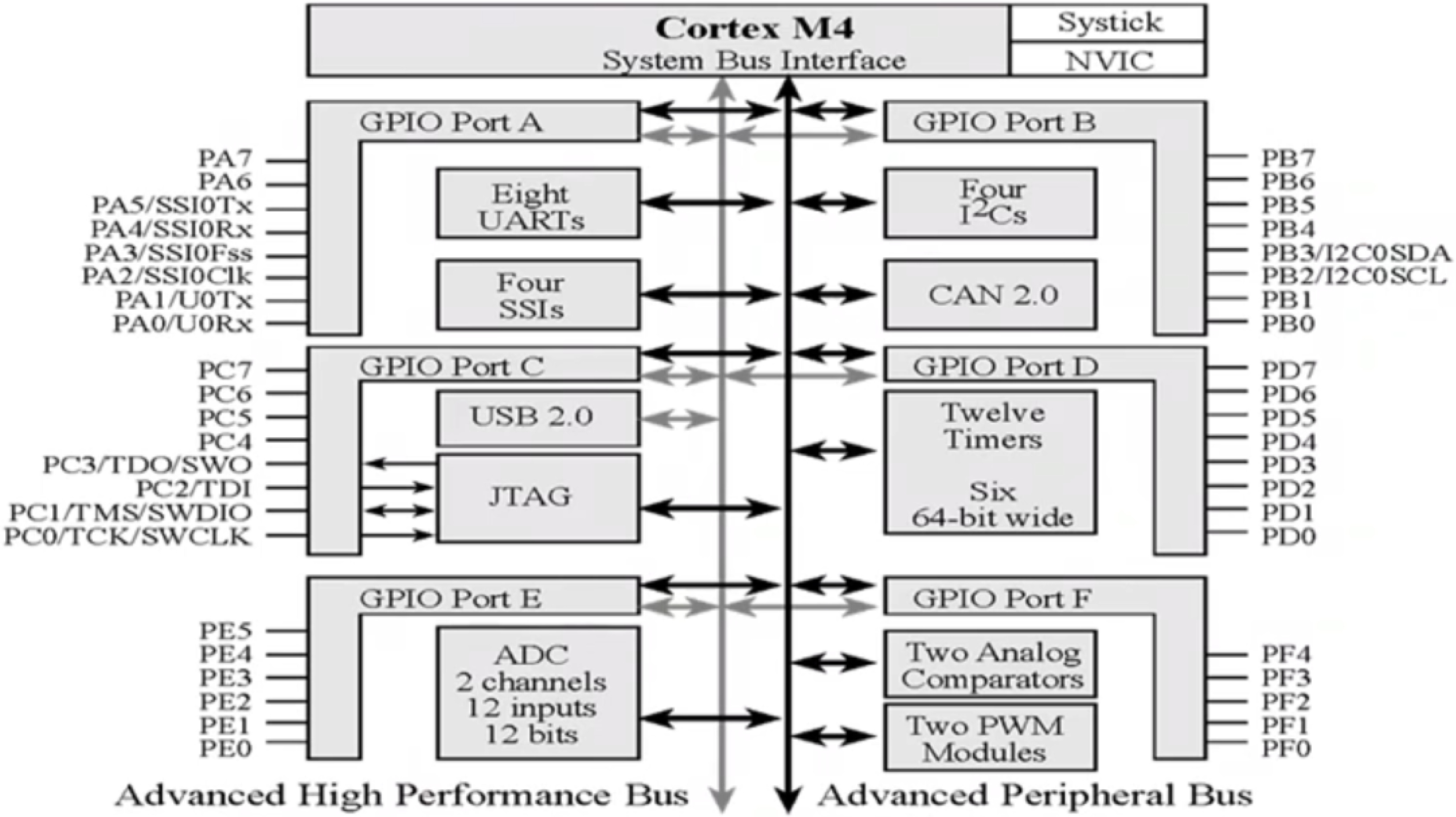


LaunchPad Switches and LEDs

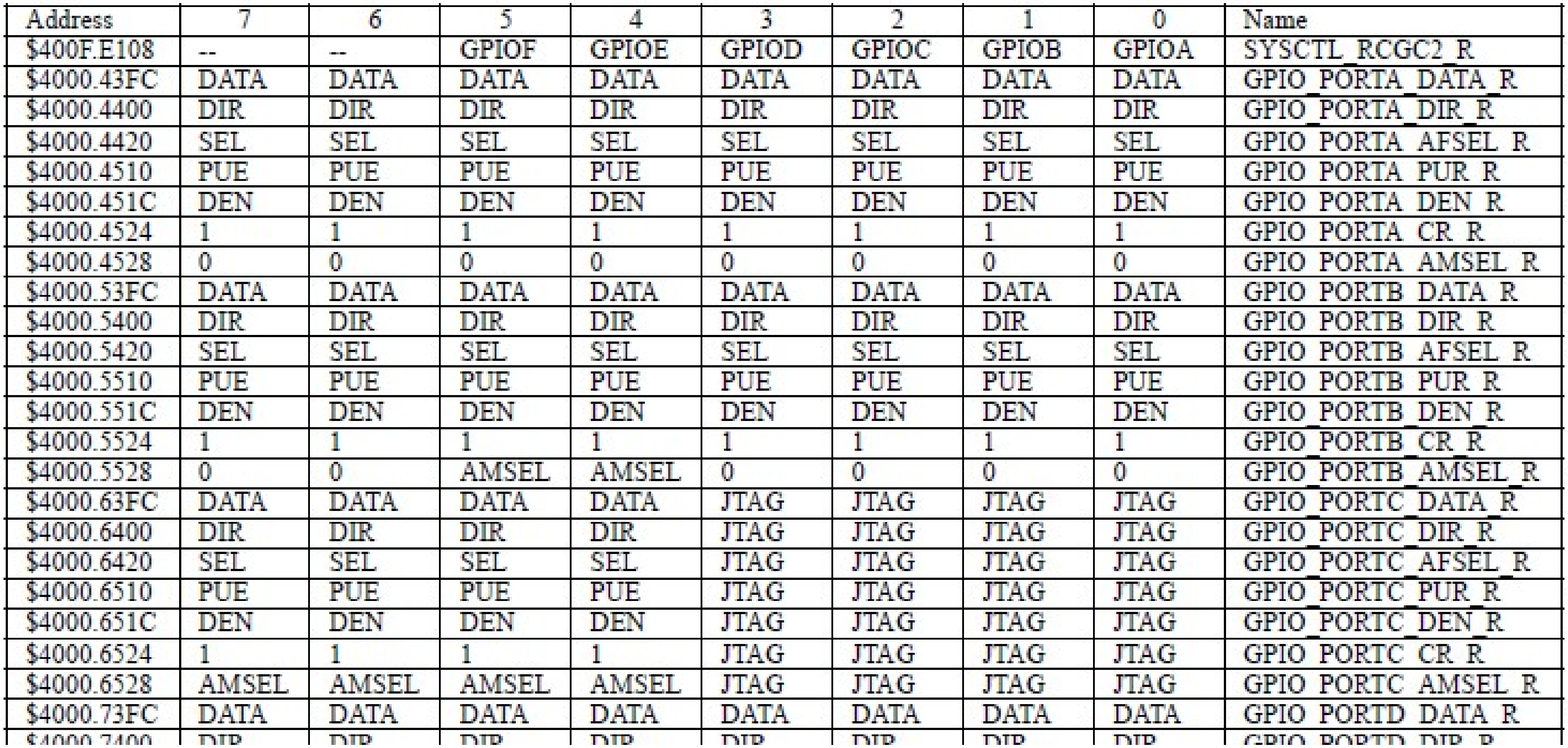


- **The switches on the LaunchPad**
    - **Negative logic**
    - **Require internal pull-up (set bits in PUR)**
- **The PF3-1 LEDs are positive logic**

Texas Instruments TM4C123



I/O Programming



Address	7	6	5	4	3	2	1	0	Name
$400F.E108	--	--	GPIOF	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	SYSCTL_RCGC2_R
$4000.43FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTA_DATA_R
$4000.4400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTA_DIR_R
$4000.4420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTA_AFSEL_R
$4000.4510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTA_PUR_R
$4000.451C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTA_DEN_R
$4000.4524	1	1	1	1	1	1	1	1	GPIO_PORTA_CR_R
$4000.4528	0	0	0	0	0	0	0	0	GPIO_PORTA_AMSEL_R
$4000.53FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTB_DATA_R
$4000.5400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTB_DIR_R
$4000.5420	SEL	SEL	SEL	SEL	SEL	SEL	SEL	SEL	GPIO_PORTB_AFSEL_R
$4000.5510	PUE	PUE	PUE	PUE	PUE	PUE	PUE	PUE	GPIO_PORTB_PUR_R
$4000.551C	DEN	DEN	DEN	DEN	DEN	DEN	DEN	DEN	GPIO_PORTB_DEN_R
$4000.5524	1	1	1	1	1	1	1	1	GPIO_PORTB_CR_R
$4000.5528	0	0	AMSEL	AMSEL	0	0	0	0	GPIO_PORTB_AMSEL_R
$4000.63FC	DATA	DATA	DATA	DATA	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DATA_R
$4000.6400	DIR	DIR	DIR	DIR	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DIR_R
$4000.6420	SEL	SEL	SEL	SEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AFSEL_R
$4000.6510	PUE	PUE	PUE	PUE	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_PUR_R
$4000.651C	DEN	DEN	DEN	DEN	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_DEN_R
$4000.6524	1	1	1	1	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_CR_R
$4000.6528	AMSEL	AMSEL	AMSEL	AMSEL	JTAG	JTAG	JTAG	JTAG	GPIO_PORTC_AMSEL_R
$4000.73FC	DATA	DATA	DATA	DATA	DATA	DATA	DATA	DATA	GPIO_PORTD_DATA_R
$4000.7400	DIR	DIR	DIR	DIR	DIR	DIR	DIR	DIR	GPIO_PORTD_DIR_R

I/O Programming & Direction Register

GPIO_PORTF_DIR_R	**Which pins are input or output.**
GPIO_PORTF_AFSEL_R	**Activate the alternate functions**
GPIO_PORTF_DEN_R:	**Digital port**
GPIO_PORTF_DATA_R	**Perform input/output on the port.**

PORTA

```
GPIO_PORTA_DATA_R    EQU  0x400043FC
GPIO_PORTA_DIR_R     EQU  0x40004400
```

Set Port Direction & Port Type

```
LDR     R1,= GPIO_PORTF_DIR_R
MOV      R0,#0x0E
STR      R0,[R1]

LDR     R1,=GPIO_PORTF_DEN_R
MOV     R0,#0xFF
STR     R0,[R1]
```

Set Port Direction & Port Type

```
GPIO_PORTF_DIR_R = 0x0E;          // PF4,PF0 in, PF3-1 out
GPIO_PORTF_AFSEL_R = 0x00;        // disable alt funct on PF7-0
GPIO_PORTF_DEN_R = 0x1F;          // enable digital I/O on PF4-0
```

I/O Ports and Control Registers

Address	7	6	5	4	3	2	1	0	Name
400F.E608	-	-	**GPIOF**	GPIOE	GPIOD	GPIOC	GPIOB	GPIOA	SYSCTL_RCGCGPIO_R
4002.53FC	-	-	-	DATA	DATA	DATA	DATA	DATA	GPIO_PORTF_DATA_R
4002.5400	-	-	-	DIR	DIR	DIR	DIR	DIR	GPIO_PORTF_DIR_R
4002.551C	-	-	-	DEN	DEN	DEN	DEN	DEN	GPIO_PORTF_DEN_R

- **Initialization (executed once at beginning)**
  1. Write *DIR* bit, 1 for output or 0 for input
  2. Set *DEN* bits to 1 to enable data pins
- **Input/output from pin**
  Input: Read from **`GPIO_PORTF_DATA_R`**
  Output: Write **`GPIO_PORTF_DATA_R`**

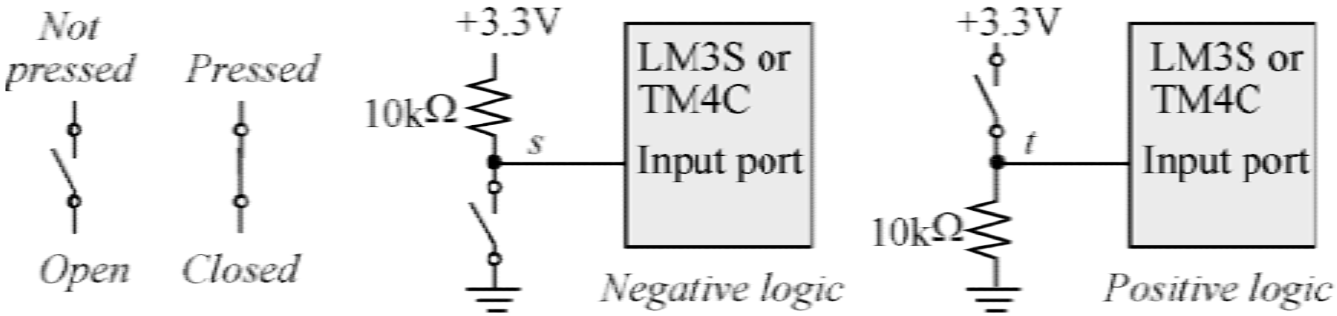
Port F LED Programming

```
DR R1, =GPIO_PORTF_DIR_R       ; R1 -> GPIO_PORTE_DIR_R
MOV R0, #0X0E                    ;PF0 , PF4 in and PF3-1 out
STR R0, [R1]                    ; set direction register

LDR R1, =GPIO_PORTF_DEN_R       ; R1 -> GPIO_PORTE_DEN_R
MOV R0, R0, #0xFF               ; enable digital port
STR R0, [R1]                    ; set digital enable register

LDR R1, =GPIO_PORTF_DATA_R
MOV R0,#0x02
STR R0,[R1]
```

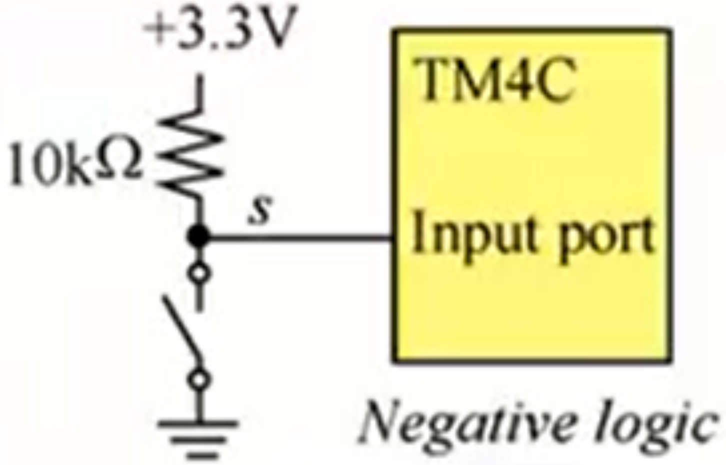
Switch Interfacing



Assembly:

```
LDR   R1,=GPIO_PORTF_DATA_R
LDR   R0,[R1]        ; read port F
AND   R0,R0,#0x11    ; PF4-PF0
```

Pull Up resistor – Switch Interface



Negative logic

Programming with C

```
#define GPIO_PORTC_DATA_R (*((volatile uint32_t *)0x400063FC))
#define GPIO_PORTC_DIR_R (*((volatile uint32_t *)0x40006400))
```

I/O Programming

```
#include "inc/tm4c123gh6pm.h"
void PortF_Init(void){
    SYSCTL_RCGCGPIO_R |= 0x00000020; // activate Port F
    while((SYSCTL_PRGPIO_R&0x00000020) == 0){};
    GPIO_PORTF_DIR_R = 0x0E; // PF4,PF0 in, PF3-1 out
    GPIO_PORTF_PUR_R = 0x11; // pull-up on PF0 and PF4
    GPIO_PORTF_DEN_R = 0x1F; // digital I/O on PF4-0
}
uint32_t PortF_Input(void){
    return (GPIO_PORTF_DATA_R&0x11); // read PF4,PF0
    inputs
}
void PortF_Output(uint32_t data){ // write PF3-PF1 outputs
    GPIO_PORTF_DATA_R = data;
}
```