Sunbeam Institute of Information Technology
Pune and Karad

Module - Micro controller Programming and Interfacing

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

12-15

regr : 0x0000hA00

```
                                  15 14 13 12
0000 0000 0000 0000 0100 1010 0000 0000
0000 0000 0000 0000 1111 0000 0000 0000
|
0000 0000 0000 0000 1111 1010 0000 0000

0000 0000 0000 0000 0001 0000 0000 0000
0000 0000 0000 0000 0010 0000 0000 0000
0000 0000 0000 0000 0100 0000 0000 0000
0000 0000 0000 0000 1000 0000 0000 0000
|
0000 0000 0000 0000 1111 0000 0000 0000
```

regr = regr | BV(12) | BV(13) | BV(14) | BV(15)

or

regr |= BV(12) | BV(13) | BV(14) | BV(15)

17-20

0x000hA000

```
                                  20 19 18 17
0000 0000 0000 0100 1010 0000 0000 0000
1111 1111 1110 0001 1111 1111 1111 1111
&
0000 0000 0000 0000 1010 0000 0000 0000

0000 0000 0000 0010 0000 0000 0000 0000
0000 0000 0000 0100 0000 0000 0000 0000
0000 0000 0000 1000 0000 0000 0000 0000
0000 0000 0001 0000 0000 0000 0000 0000
|
0000 0000 0001 1110 0000 0000 0000 0000
~
1111 1111 1110 0001 1111 1111 1111 1111
```

regr = regr & ~(BV(17) | BV(18) | BV(19) | BV(20))

or

regr &= ~(BV(17) | BV(18) | BV(19) | BV(20))

19 - 24

0x104A0000

24 23 22 21 20 19

0001 0000 0100 1010 0000 0000 0000 0000

>>19

0000 0000 0000 0000 0000 0010 0000 1001

0000 0000 0000 0000 0000 0000 0011 1111

&

0000 0000 0000 0000 0000 0000 0000 1001

value = (regr >> 19) & 0x0000003F

0001 0000 0100 1010 0000 0000 0000 0000

0001 0001 1111 1000 0000 0000 0000 0000

0000 0000 0100 1000 0000 0000 0000 0000

>>19

0000 0000 0000 0000 0000 0000 0000 1001

8 - 15

0x52

0x1004A000

15 14 13 12 11 10 9 8

0001 0000 0000 0100 1010 0000 0000 0000

0001 0000 0000 0100 0000 0000 0000 0000

0000 0000 0000 0000 0101 0010 0000 0000

|

0001 0000 0000 0100 0101 0010 0000 0000

regr &= ~(BV(8) | BV(9) | BV(10) | BV(11) | BV(12) | BV(13) | BV(14) | BV(15))

regr |= (value << 8)

value = 0x52

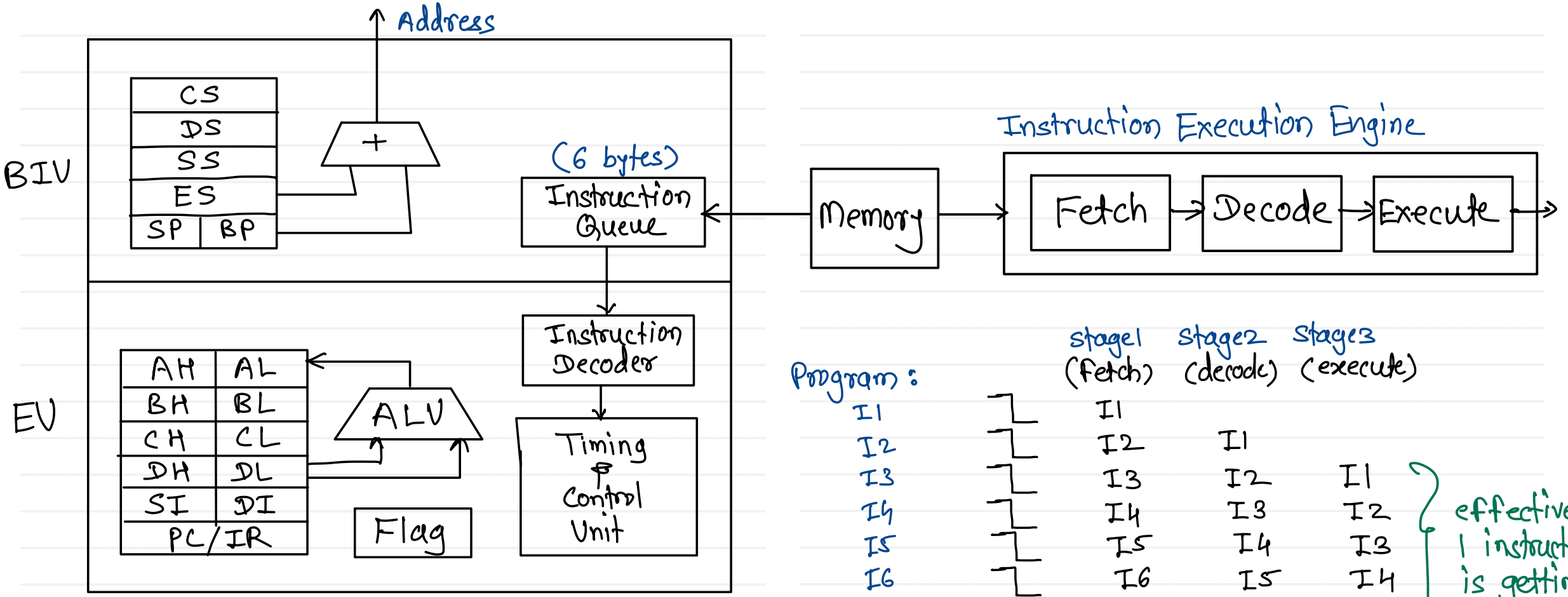
value = 0000 0000 0000 0000 0000 0000 0101 0010

value<<8 = 0000 0000 0000 0000 0101 0010 0000 0000

```
regx = 0001 0000 0000 0100 1010 0000 0000 0000
     | 0000 0000 0000 0000 0101 0010 0000 0000
-------------------------------------------------
       0001 0000 0000 0100 1111 0010 0000 0000
```

```
regx = 0001 0000 0000 0100 0000 0000 0000 0000
     | 0000 0000 0000 0000 0101 0010 0000 0000
-------------------------------------------------
       0001 0000 0000 0100 0101 0010 0000 0000
```

Instruction queue vs Instruction pipeline



Program :

I1
I2
I3
I4
I5
I6

	stage1 (fetch)	stage2 (decode)	stage3 (execute)
	I1		
	I2	I1	
	I3	I2	I1
	I4	I3	I2
	I5	I4	I3
	I6	I5	I4

effectively 1 instruction is getting completed per CPU cycle

Instruction pipeline hazards

Control Hazard

```
1        MOV A, 10
2        MOV B, 20
3        CMP
4        Branch → T
5        B is greater
6        JUMP E
7    T:  A is greater
8    E:
```

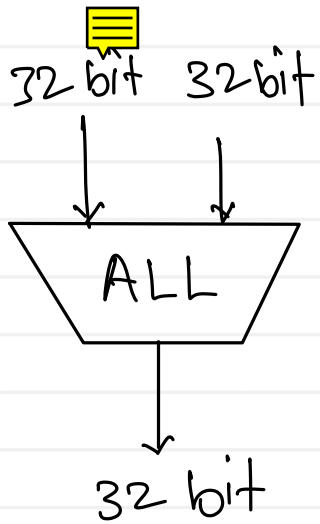
	F	D	E
1	1		
2	2	1	
3	3	2	1
4	4	3	2
5	5	4	3
6	6	5	4
7	7		-
8	8	7	-
		8	7
			8

Data Hazard

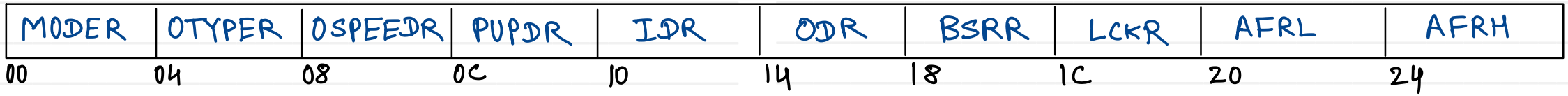
```
1    MOV A, 10
2    MOV B, 20
3    MUL A, B
4    STA 0x12345678
5    ....
```

	F	D	E
1	1		
2	2	1	
3	3	2	1
4	4	3	2
5	5	4	3
			3
			3
		5	4
			5

Pipeline is halted



GPIO registers



start:
0x4002 0C00

end:
0x4002 0FFF

(struct GPIO *)0x4002 0C00 -> .....

```
struct GPIO {
        uint32_t MODER;
        uint32_t OTYPER;
        uint32_t OSPEEDR;
        uint32_t PUPDR;
        uint32_t IDR;
        uint32_t ODR;
        uint32_t BSRR;
        uint32_t LCKR;
        uint32_t AFRL;
        uint32_t AFRH;
};
```

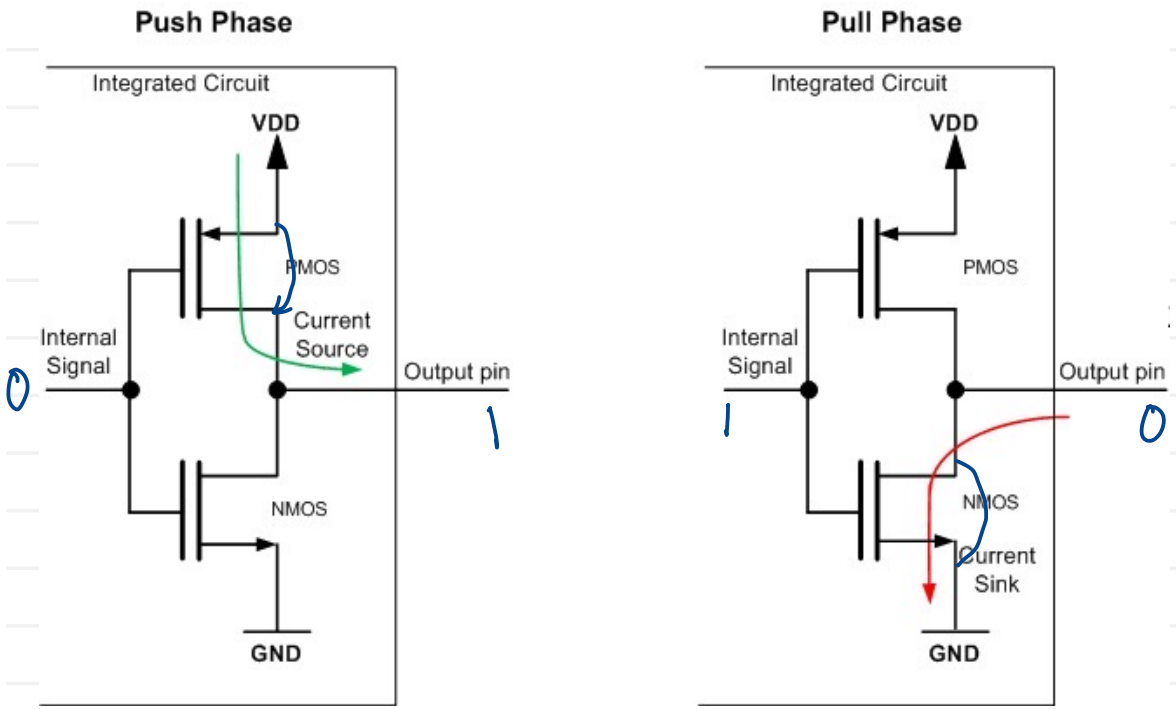
#define GPIOD_Base 0x4002 0C00

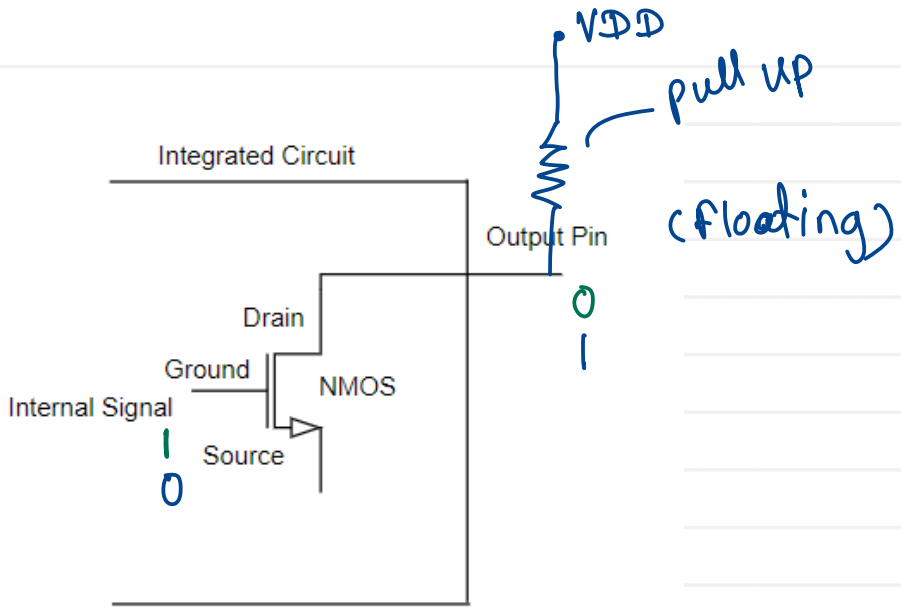
(struct GPIO *) GPIOD_Base -> .....

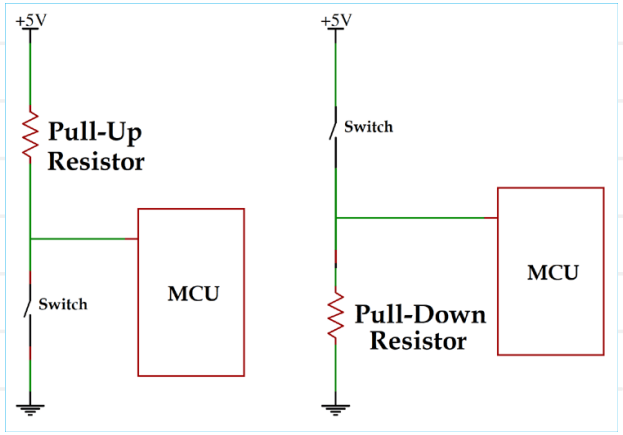
#define (struct GPIO*)GPIOD_Base GPIOD

GPIOD -> . . . .

Push-pull vs Open drain configuration







Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com