Practice Sheet

Q1. Suppose, you need to configure GPIO Port B’s pin 3 as output port. Write the registers that you need and the operation you need to execute for interfacing.

Sol.

We need 2 registers. RCC register for activating the GPIO port & MODER register to configure the pin as output port.

Operation:

RCC -> 1<<1 ; [Since port B enable is at 1 no. index and we need 1 bit left shift operation]
MODER -> 1<<6 ; [Since Pin 3 starts from 6 no. index [to configure 1 pin in MODER register we need 2 slots] and output mode = 1 [or 01 in binary]]

Q2. Suppose, you need to configure GPIO Port A’s pin 5 as input port. Write the registers that you need and the operation you need to execute for interfacing.

Sol.

We need 2 registers. RCC register for activating the GPIO port & MODER register to configure the pin as output port.

Operation:

RCC -> 1<<0 ; [Since port A enable is at 0 no. index and we need 0 bit left shift operation/ no left shift operation]
MODER -> 0<<10 ; [Since Pin 5 starts from 10 no. index [to configure 1 pin in MODER register we need 2 slots] and input mode = 0 [or 00 in binary]]

Q3. Suppose, you need to configure GPIO Port D’s pin 3 as output port. You need to send the data at “Fast Speed” and your output type should follow an open drain system. Write the registers that you need and the operation you need to execute for interfacing.

Sol.

We need 4 registers.

1. **RCC register** for activating the GPIO port [**configure GPIO Port D**],
2. **MODER register** [**pin 3 as output port**] to configure the pin as output port
3. **OSPEEDER register** to configure speed [**“Fast Speed”**]
4. **OTYPER register** to configure output type [**Type**]

Operation:

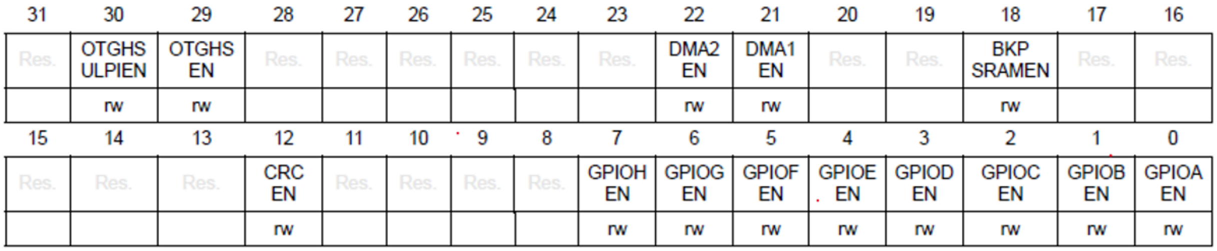
RCC -> 1<<3 ; [Since port D enable is at 3 no. index and we need 3 bit left shift operation]
MODER -> 1<<6 ; [01/1 = Output mode, Position 6 = pin 3]
OSPEEDER -> 2<<6; [10/2 = Fast speed, Position 6 = pin 3]
OTYPER -> 1<<3; [1 = Opendrain, Position = pin 3]

Q4. RCC -> AHB1EN |= (1<<2) ; // To enable/ activate GPIOx
RCC -> AHB1EN |= (1<<3); // To enable/ activate GPIOy
GPIOx ->MODER |= 0<<0; // Setting up the mode of pins
GPIOy -> MODER |= 1<<2; // Setting up the mode of pins

i) Identify which GPIO ports are we enabling for communication.
ii) Identify which pin are we enabling of GPIOx? State the mode of the pin as well.
iii) Identify which pin are we enabling of GPIOy? State the mode of the pin as well.

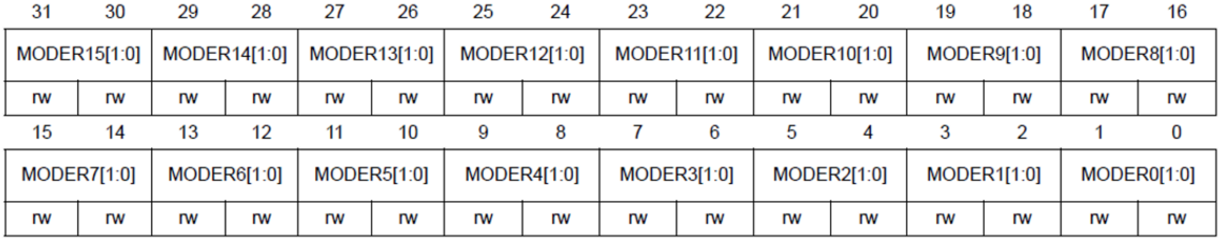
<u>Sol.</u>

RCC Register:



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPIEN	OTGHS EN	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 EN	DMA1 EN	Res.	Res.	BKP SRAMEN	Res.	Res.
	rw	rw							rw	rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	Res.	Res.	Res.	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw	rw	rw	rw	rw	rw	rw	rw

MODER Register:



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

i) GPIOX = GPIOC and GPIOY = GPIOD
ii) GPIOX = GPIOC, we are enabling the PIN0 of port C and the mode of PIN0 is configured as intput pin.
iii) GPIOY = GPIOD, we are enabling the PIN1 of port D and the mode of PIN1 is configured as output pin.

Q5. RCC -> AHB1EN |= (1<<1) ;
RCC -> AHB1EN |= (1<<3);

RCC -> AHB1EN &= (1<<4) ;

After executing the commands what will be the which port will be enabled?

Sol.

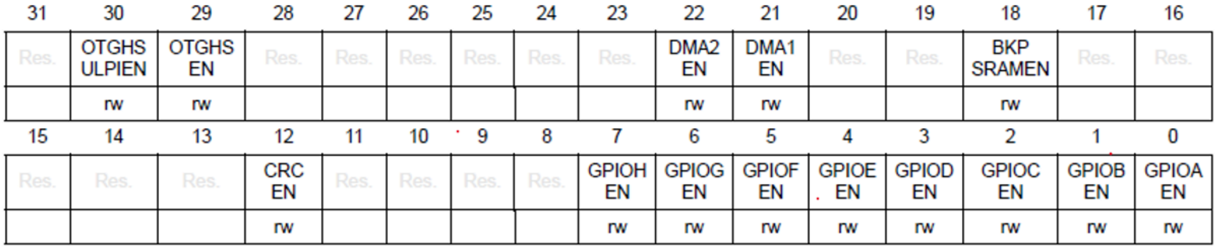
GPIO port E.

--

|= : OR operation

&= : AND operation

RCC Register:



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	OTGHS ULPIEN	OTGHS EN	Res.	Res.	Res.	Res.	Res.	Res.	DMA2 EN	DMA1 EN	Res.	Res.	BKP SRAMEN	Res.	Res.
	rw	rw							rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRC EN	Res.	Res.	Res.	Res.	GPIOH EN	GPIOG EN	GPIOF EN	GPIOE EN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw					rw	rw	rw	rw	rw	rw	rw	rw

Operation:

RCC -> AHB1EN |= (1<<1) ;

RCC -> AHB1EN |= (1<<3);

RCC -> AHB1EN &= 1<<4 ;