

PSI2662 – Projeto em Sistemas Eletrônicos Embarcados: Sensores e Atuadores

Portas de Entrada e Saída de Uso Geral (General Purpose I/O) e seu endereço

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – grehder@lme.usp.br



Segundo Semestre de 2015



Mapa da Memória do Cortex M0+

Memória do ARM é mapeada em 4 Gb

KL25Z

	Allocated size	Allocated address
Flash	128KB	0x00000000 to 0x0001FFFF
SRAM	16KB	0x1FFFF000 to 0x20002FFF
I/O	All the peripherals	0x400FF000 to 0x400FFFFF

Table 2-1: Memory Map in KL25Z128VLK4

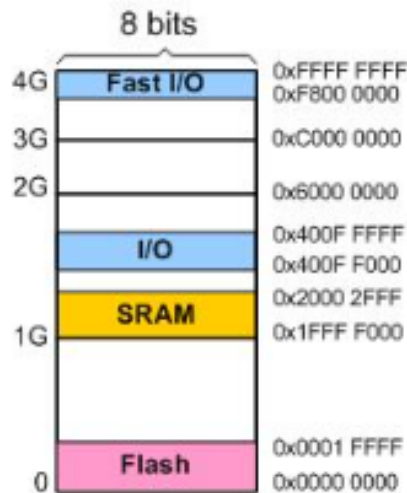


Figure 2-3: Memory Map

- Flash – Código do Programa e dados fixos como tabelas no ROM
- SRAM – variáveis
- Periféricos - endereços para registradores associados a I/Os, Timers, ADCs etc.

Olhar página 177 do KL25 Sub-Family Reference Manual para definições do registrador



General Purpose I/O (GPIO)

- 5 portas (A, B, C, D e E);
- 32 pinos por porta (PTA0 .. PTA31; PTB0 .. PTB21 etc.);
- Nem todos pinos são implementados;
- Chips da ARM têm dois barramentos: APB (Advanced Peripheral Bus) e o AHB (Advanced High-Performance Bus);
- AHB muito mais rápido que o APB.

Enderaçamento

- GPIO Port A (APB): 0x400F F000
- GPIO Port B (APB): 0x400F F040
- GPIO Port C (APB): 0x400F F080
- GPIO Port D (APB): 0x400F F0C0
- GPIO Port E (APB): 0x400F F100

- GPIO Port A (AHB): 0xF80F F000
- GPIO Port B (AHB): 0xF80F F040
- GPIO Port C (AHB): 0xF80F F080
- GPIO Port D (AHB): 0xF80F F0C0
- GPIO Port E (AHB): 0xF80F F100

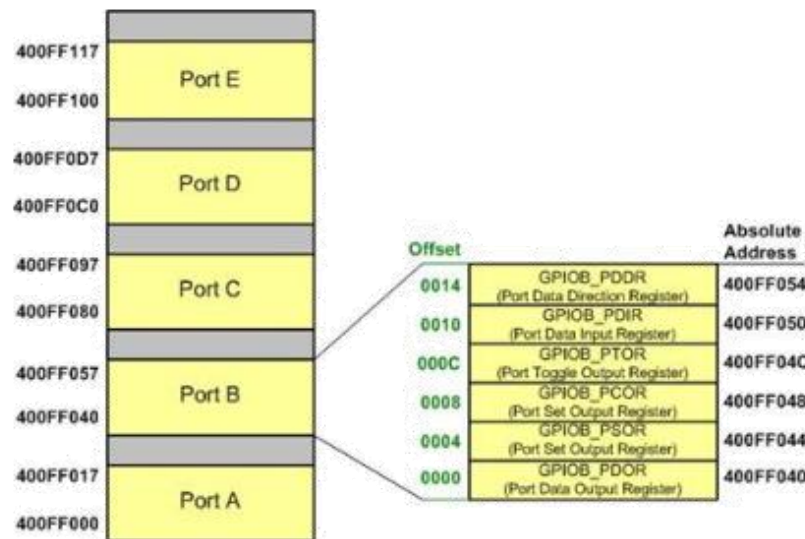
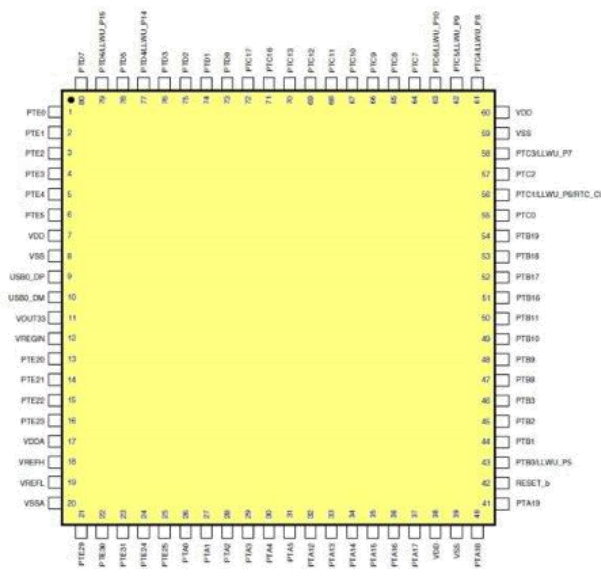


Figure 2-4: GPIO Memory Map



Registadores das GPIO

- Existem dois registradores associados a cada porta:
 - Direction Register – define se o pino é uma entrada ou saída;
 - Data Register – escreve ou lê dados do pino

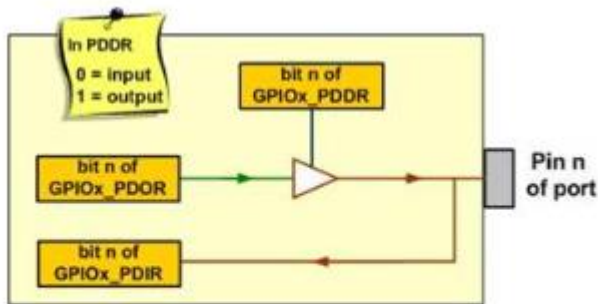


Figure 2-7: The Data and Direction Registers and a Simplified View of an I/O pin

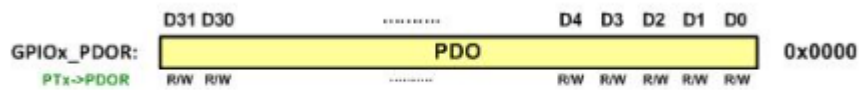


Figure 2-8: GPIOx_PDOR (Port Data Output Register)

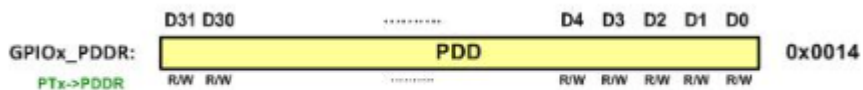


Figure 2-9: GPIOx_PDDR (Port Data Direction Register)

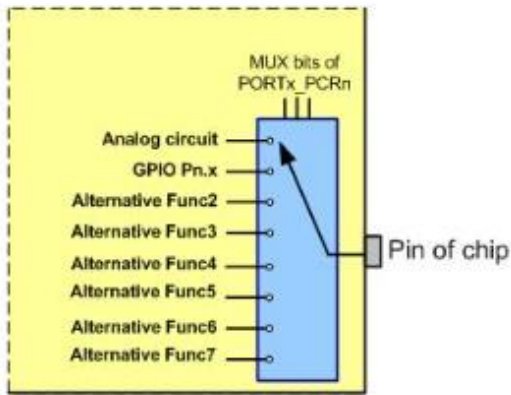
Address	Name	Description	Type	Reset Value
0x400F000	GPIOA_PDOR	Port Data Output Register	R/W	0x00000000
0x400F004	GPIOA_PSOR	Port Set Output register	W (always reads 0)	0x00000000
0x400F008	GPIOA_PCOR	Port Clear Output Register	W (always reads 0)	0x00000000
0x400F00C	GPIOA_PTOR	Port Toggle Output Register	W (always reads 0)	0x00000000
0x400F010	GPIOA_PDIR	Port Data Input Register	R	0x00000000
0x400F014	GPIOA_PDDR	Port Data Direction Register	R/W	0x00000000

Table 2-3: Some GPIO Registers for PORTA



Funções alternativas dos pinos

- Pin multiplexing
- Função controlado pelo PORTx_PCRn (Port x Pin n Control Register)
- Bits mais importantes: D10-D8 (Mux Control)



Address: Base address + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
R	0							ISF	0				IROC				
W								w1c									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R	0					MUX			0	DSE	0	PFE	0	SRE	PE	PS	
W						MUX											
Reset	0	0	0	0	0	x*	x*	x*	0	x*	0	x*	0	x*	x*	x*	

* Notes:
• x = Undefined at reset.

Olhar página 183 do KL25 Sub-Family Reference Manual para definições do registrador

Figure 2-10: Alternative Functions of Pins

BIT	Field	Description
0	Pull Select (PS)	If the PE field is set, the field chooses between pull-up and pull-down resistors. 0: pull-down resistor, 1: pull-up resistor
1	Pull Enable (PE)	0: Disable the internal pull resistors 1: Enable the internal pull resistors
2	Slew Rate Enable (SRE)	0: Fast slew rate 1: Slow slew rate
4	Passive Filter Enable (PFE)	0: Passive input filter is disabled 1: Passive input filter is enabled
6	Drive Strength Enable (DSE)	0: Low drive strength 1: High drive strength
10-8	Pin Mux Control (MUX)	

Pin Mux Control

- 000 Pin disabled (analog).
- 001 Alternative 1 (GPIO).
- 010 Alternative 2 (chip-specific).
- 011 Alternative 3 (chip-specific).
- 100 Alternative 4 (chip-specific).
- 101 Alternative 5 (chip-specific).
- 110 Alternative 6 (chip-specific).
- 111 Alternative 7 (chip-specific).



Exemplo

- Configurar Pinos PTB18 e PTB19 para output:
 - $\text{GPIOB_PDDR} = 0x000C\ 0000 \rightarrow 0b0000\ 0000\ 0000\ 1100\ 0000\ 0000\ 0000\ 0000$

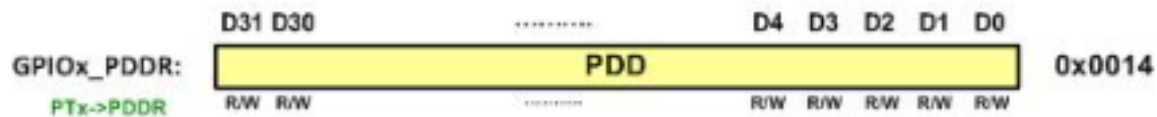


Figure 2-9: GPIOx_PDDR (Port Data Direction Register)

- Endereço do registrador GPIOB_PDDR:
 - Para Porta B = 0x4000 F040
 - Offset de 0x0014
 - Endereço do 0x4000 F054

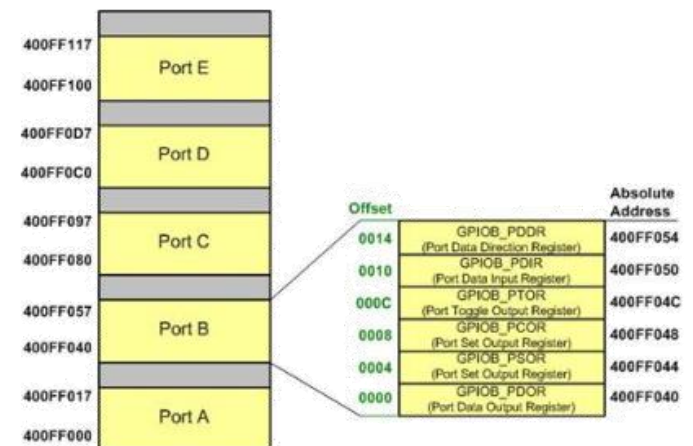


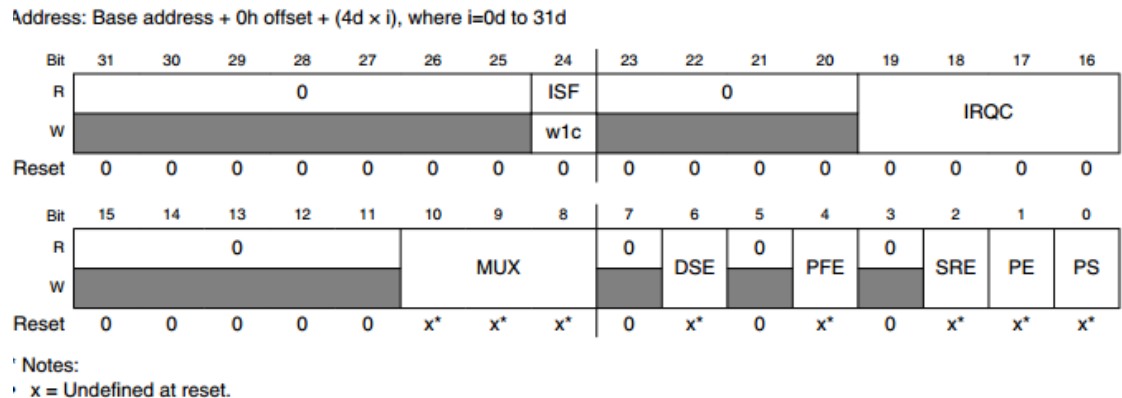
Figure 2-4: GPIO Memory Map



Exemplo

- Configurar Pino PTB18 com slow slew rate, high drive e sem pull-up:
 - Registrado PORTB_PCR18:
 - 0b0000 0000 0000 0000 0000 0001 0100 0100
 - 0x0000 0144
- Endereço:
 - PORTB_PCR18 = 4004 A048

BIT	Field	Description
0	Pull Select (PS)	If the PE field is set, the field chooses between pull-up and pull-down resistors. 0: pull-down resistor, 1: pull-up resistor
1	Pull Enable (PE)	0: Disable the internal pull resistors 1: Enable the internal pull resistors
2	Slew Rate Enable (SRE)	0: Fast slew rate 1: Slow slew rate
4	Passive Filter Enable (PFE)	0: Passive input filter is disabled 1: Passive input filter is enabled
6	Drive Strength Enable (DSE)	0: Low drive strength 1: High drive strength
10-8	Pin Mux Control (MUX)	



Olhar página 177 do KL25 Sub-Family Reference Manual para definições do registrador



Clock para GPIO

- O clock deve ser habilitado antes de configurar a porta;
- Registrador SIM_SCGC5 habilita o clock para todas as portas;
- Para economizar energia o clock das portas que não são utilizadas não devem ser habilitado;
- Endereço do SIM_SCGC5:
 - $0x4004\ 7000 + 0x1038 = 0x4004\ 8038$

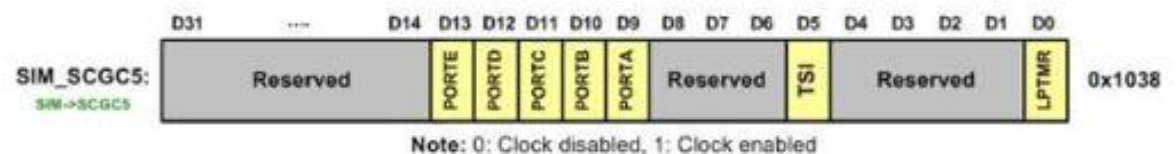


Figure 2-13: SIM_SCGC5 (System Clock Gating Control Register 5) Register



Exemplo

- Habilitar clock somente para porta B:

- $SIM_SCGC5 = 0b0000\ 0000\ 0000\ 0000\ 0000\ 0100\ 0000\ 0000$
- $SIM_SCGC5 = 0x0000\ 0400$
- $SIM_SCGC5 |= 0x0400$ (OR binário - *bitwise*)
 - $SIM_SCGC5 = SIM+SCGC5 | 0x0400$ (seta somente o bit de interesse)

- Operações Booleanas Binárias:

- OR | $0x04 | 0x68 = 0x6C$
- AND & $0x35 \& 0x0F = 0x05$
- XOR ^ $0x54 | 0x78 = 0x2C$
- Invert ~ $\sim 0x55 = 0xAA$
- Shift << ou >>
 $0b0001\ 0000 \gg 3 = 0b0000\ 0010$
 $1 \ll 3 = 0b0000\ 1000$

Qualquer número | 1 = 1

Qualquer número | 0 = sem mudança

Qualquer número & 1 = sem mudança

Qualquer número & 0 = 0

Qualquer número ^ 1 = complemento

Qualquer número | 0 = sem mudança



Exercício

- Objetivo: Piscar LED Verde com 2 segundo de período
 - Fazer no programa Codewarrior (C:\Freescale\Eclipse\cwide.exe)
 - Sequência do programa
 - (1) Habilitar clock da porta B;
 - (2) Configurar Pino 19 (Pin Control Register);
 - (3) Setar a direção do Pino;
 - (4) Habilitar saída;
 - (5) Função de espera;
 - (6) Desabilitar saída;
 - (7) Função de espera;
 - Repetir passos (4)-(7).



Definir Endereço dos Registradores

Exemplo

```
/* Define o endereço do registrador SIM_SCGC5 */  
#define SIM_SCGC5 (*((volatile unsigned int*)0x40048038))
```

Função de espera

```
/* no main */
```

```
void delayMs (int n);
```

```
/* Função: Espera n milisegundos */
```

```
/* esta função depende do clock default do microcontrolador. Para o KL25Z a frequência é  
21 MHz aproximadamente. O valor do contador deverá ser ajustado para se conseguir o  
tempo de espera desejado. */
```

```
Void delayMs (int n) {  
    int i;  
    int j;  
    for (i = 0; i < n; i++)  
        for (j = 0; j < 7000; j++) {}  
}
```