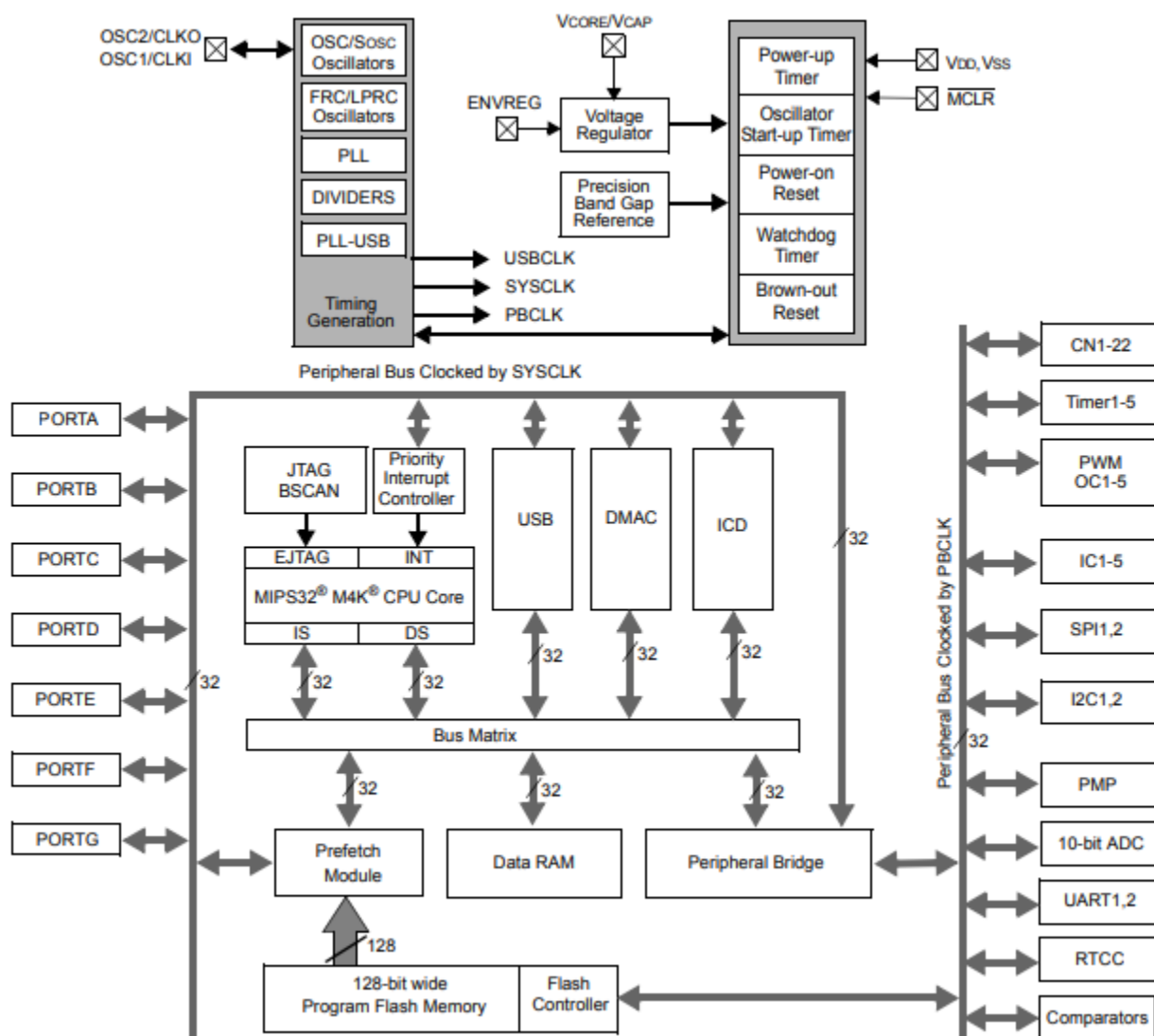


Laborator 6

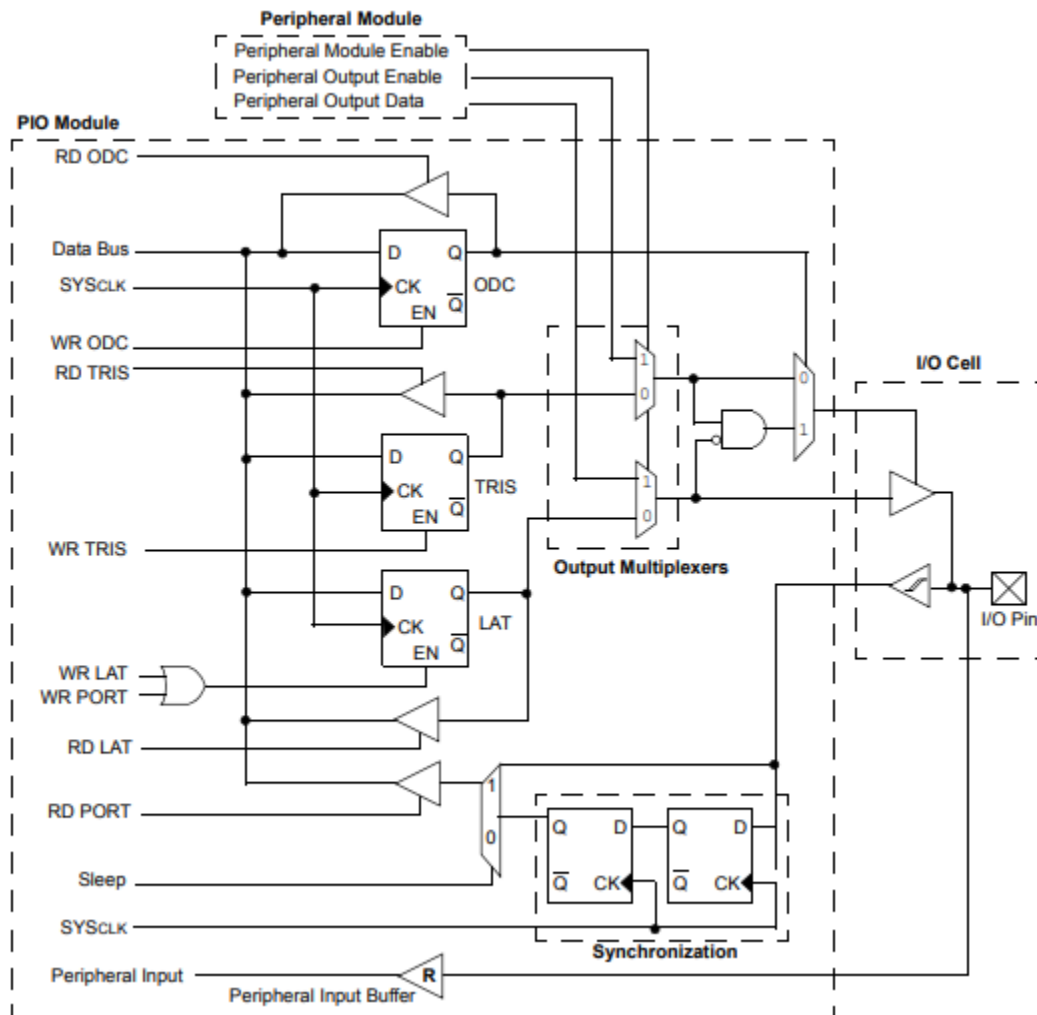
Pinii I/O de uz general sunt cei mai simpli dintre periferice. Acestea permit microcontroller-ului să monitorizeze și să controleze alte dispozitive (ex: led-uri și butoane). Pentru a adăuga flexibilitate și funcționalitate, unii pinii sunt multiplexați cu o funcție alternativă. Aceste funcții depind de caracteristicile perifericelor dispozitivului. În general, când funcționează un periferic, acel pin nu poate fi utilizat ca pin I / O cu scop general.

6.1 Diagrama bloc pentru familia de device-uri PIC32



Așa cum se poate vedea din imagine, in cazul microcontrollere-lor PIC32 exista 7 PORT-uri I/O: PORTA, PORTB, PORTC, PORTD, PORTE si PORTF de cate 16 biți.

6.2 Structura interna a unui PORT



6.3 Registrii unui port

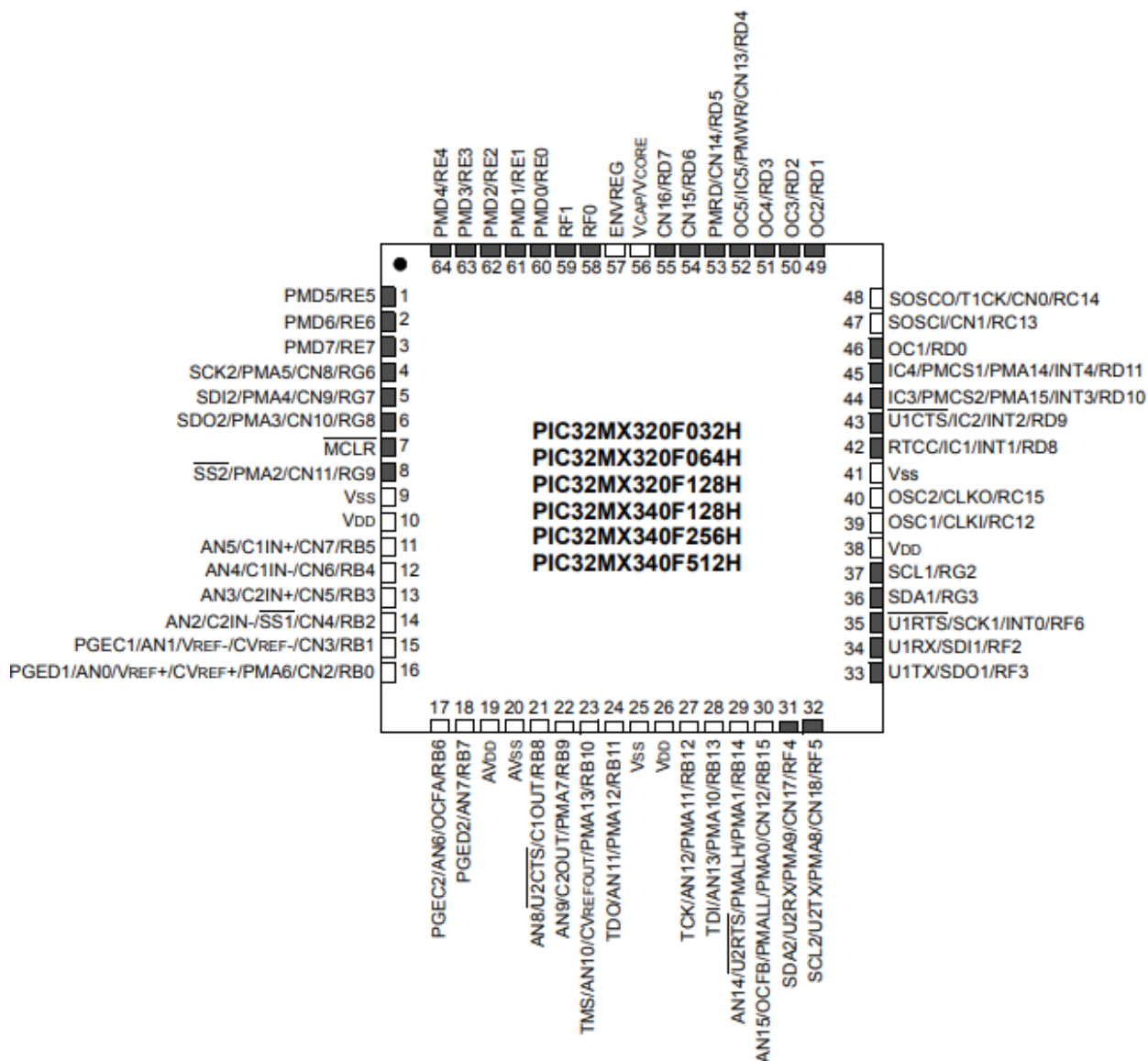
Perifericul PORT are trei registrii prezentați in tabelul de mai jos:

Nume	Descriere
TRIS	<p>tri-state control register determina daca un pin este configurat ca pin de intrare sau pin de ieșire.</p> <p>Toți pinii sunt definiți ca input după aplicarea unui reset, însă aceștia pot fi configurați din software:</p> <ul style="list-style-type: none"> - TRIS = 1, pin-ul este configurat ca input - TRIS = 0, pin-ul este configurat ca output

PORT	Registru folosit pentru a citi starea curenta a pin-ului I/O. O scriere la acest registru este transformata intr-o scriere la registrul LAT.
LAT	Registrul este utilizat pentru a scrie o valoare la pinii I/O.

6.4 Pinii familiei de device-uri PIC32 si corespondenta cu placa de dezvoltare Arduino uc32

Pinii device-ului PIC32 sunt prezentați in figura de mai jos.



Pinii I/O ai porturilor sunt notati RBx, RCx, RDx, REx, RFx sau RGx, unde x este o valoare de la 0-15.

În cazul plăcii de dezvoltare Arduino uc32 o parte din pinii porturilor sunt conectați la porturile digitale JP6 și JP5 conform tabelului prezentat în manualul de referință.

Pin #	Connector Pin #	PIC32 Pin #	PIC32 Signal	Notes
26	J6-02	60	PMD0/RE0	
27	J6-04	61	PMD1/RE1	
28	J6-06	62	PMD2/RE2	
29	J6-08	63	PMD3/RE3	
30	J6-10	64	PMD4/RE4	
31	J6-12	1	PMD5/RE5	
32	J6-14	2	PMD6/RE6	
33	J6-16	3	PMD7/RE7	
34	J5-02	53	PMRD/CN14/RD5	
35	J5-04	45	IC4/PMCS1/PMA14/INT4/RD11	
36	J5-06	54	CN15/RD6	
37	J5-08	55	CN16/RD7	
38	J5-10	35	U1RTS/BCLK1/SCK1/INT0/RF6	
39	J5-12	31	PMA9/U2RX/SDA2/CN17/RF4	
40	J5-14	32	PMA8/U2TX/SCL2/CN18/RF5	
41	J5-16	15	PGC1/AN1/VREF-/CVREF-/CN3/RB1	

Tabelul în care sunt prezentați toți pinii pentru conectorii digitali:

https://reference.digilentinc.com/_media/reference/microprocessor/uc32/uc32_rm.pdf

6.5 Exerciții

1. Rulați programul următor care aprinde un led conectat la pinul 0 al portului E.

```
.data
    TRISE_VADDR: .word 0xBF886100
    TRISE_VALUE: .word 0xFFFFFFFF
    PORTE_VADDR: .word 0xBF886110
.text
.globl main

main:
    #set RE0 as output
    lw $t0, TRISE_VADDR
    lw $t1, TRISE_VALUE
    sw $t1, 0($t0)

    #load port address
    lw $s0, PORTE_VADDR
```

```

    #turn off all the LEDs
    sw $0, 0($s0)

LOOP:
    lw $a0, 0($s0)
    ori $a0, $a0, 0b0001
    sw $a0, 0($s0)
    j LOOP

```

2. Modificați programul anterior astfel încât sa se aprindă toate cele 4 led-uri conectate la pinii 0-4 ai portului E.
3. Realizați un program care aprinde si stinge cu intermitenta un led la un interval de 1s. Țineți cont de faptul de ceasul de funcționare al microprocesorului (80MHz).

```

.data
    TRISE_VADDR: .word 0xBF886100
    TRISE_VALUE: .word 0x000000F0
    PORTE_VADDR: .word 0xBF886110
.text
.globl main

main:
    #set RE0 as output
    lw $t0, TRISE_VADDR
    lw $t1, TRISE_VALUE
    sw $t1, 0($t0)
    #load port address
    lw $s0, PORTE_VADDR
    #turn off all the LEDs
    sw $0, 0($s0)
LOOP:
    #turn all LEDs on
    lw $a0, 0($s0)
    ori $a0, $a0, 0b00001111
    sw $a0, 0($s0)
    #wait for LEDs to turn on for 0.5s
    jal DELAY
    #turn all LEDs off
    lw $a0, 0($s0)
    andi $a0, $a0, 0b11110000
    sw $a0, 0($s0)
    #wait for LEDs to turn off for 0.5s
    jal DELAY
    j LOOP

#delay subroutine
DELAY:
    li $s4, 256000 #change this value for 0.5seconds delay
DELAY_LOOP:
    addi $s4, $s4, -1
    bne $s4, $0, DELAY_LOOP

    jr $ra
    nop

```

4. Realizați un counter binar pe cele 4 led-uri.