

Porti in registri

ARM je 32 bitna arhitektura, zato je tudi večina registrov 32 bitnih, razen če so zaradi posebne funkcije krajši. Registri so naslovi v pomnilniku in do njih lahko dostopamo kot do ostalih spremenljivk v c-ju (nekateri so sicer read ali write-only). Preko njih je naš program povezan s periferijo, npr. vhodno izhodnimi pini.

Registri, ki smo si jih pogledali v tej vaji so:

PINSEL0 – po dva bita na vsak vhodni pin, s podatki v tem registru nastavljam funkcije pinov (večina pinov ima 4 možne funkcije, 00 pomeni splošni vhodno izhodni pin)

V primeru da so preko registra PINSELx pini konfigurirani kot vhodno izhodni pini lahko do njih dostopamo preko naslednjih registrov (ti imajo po en bit na pin):

IO0DIR – pove ali je pin vezan kot izhodni ali vhodni. 1 pomeni izhodni, 0 vhodni

IO0PIN – Če je vhodni ni notranje povezan (plava) in mora bit stanje nanj pripeljano od zunaj. V registru IO0PIN je pri pinih ki so konfigurirani kot vhod stanje na vhodu.

Če je pin nastavljen kot izhodni je na pinu povezano 3.3V (tok lahko teče ven) če je stanje 1 ali 0V (povezava na GND, tok lahko teče noter) če je stanje 0. Na te bite tudi lahko vplivamo s pisanjem v IO0PIN register in z IO0CLR ter IO0SET.

IO0CLR, IO0SET – registra ki sta povezana z registrom IO0PIN kot je že razvidno iz imen registrov se v registru IO0PIN pobrišejo ali postavijo biti, ki ju zapišemo v ta registra.

V vaji smo naslavljali led diode in tipke:

Led diode: p0.4-p0.7 (4.-7. bit)

Tipke: p0.12-p0.15 (12.-15.bit)

PINSEL0

Zgradba registra po bitih in biti ki nas zanimajo

00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
P0.15	P0.14	P0.13	P0.12	P0.11	P0.10	P0.9	P0.8	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

Na 0 moramo torej postaviti bite od 8-15 (za ledice) ter 24-31 (za tipke):

```
PINSEL &= 0b00000000111111110000000011111111
```

```
PINSEL &= 0x00FF00FF
```

IO0DIR

V IO0DIR želimo vpisati enice na mesta kjer so ledice (izhodi) ter nastaviti ničle na mesta kjer so tipke (vhodi), pozor tu je le en bit na pin (št. bita sovpada s številko pina):

```
// pobrišemo bite 12-15
```

```
// IO0DIR &= 0b11111111111111110000111111111111
```

```
IO0DIR &= 0xFFFF0FFF
```

```
// enako lahko dosežemo z:
```

```
IO0DIR &= ~0x0000f000;
```

```
// nastavimo bite 4-7
// IO0DIR |= 0b00000000000000000000000011110000
IO0DIR |= 0x000000F0
```

Sedaj lahko ledice prižgamo tako da pišemo v IO0PIN bite 4-7 oz. z uporabo IO0CLR in IO0SET registrov.

```
IO0SET=0x000000F0 // prižgemo ledice
IO0CLR=0x000000F0 // ugasnemo ledice
```

Lahko tudi prižgemo ali ugasnemo posamezne ledice (spremenljivka »ledice«, vrednost 0-15)

```
IO0SET= (ledice&0xF)<<4;
IO0CLR= (ledice&0xF)<<4;
// najprej z masko odstranimo morebitne odvečne bite (če je spremenljivka ledice večja od
15), potem zamaknemo za 4 bite v levo
```

Podobno beremo tipke:

```
tipke = (IO0PIN & 0x0000F000) >> 12;
// najprej z masko odstranimo nepotrebne informacije, potem pa rezultat zamaknemo za 12
bitov v desno da dobimo vrednosti od 0 do 15
// enako naredimo z:
tipke = (IO0PIN>>12) & 0xF;
```

Z maskami lahko tudi preverimo npr. če je prisoten drugi bit (! biti se štejejo od 0) oziroma tipka T2.

```
if (tipke & (1<<2)) naredi_nekaj(); // izvede se če je prisoten drugi bit
if (!(tipke & (1<<2))) naredi_nekaj(); // izvede se če ni prisoten drugi bit
if (tipke & (1<<0)) naredi_nekaj(); // izvede se če je prisoten ničti bit
```

Vrednosti določenega pina lahko zamenjamo / invertiramo s XOR operacijo:

```
IO0PIN ^= 0x000000f0; // zamenjamo stanje ledic
// pri pisanju seveda lahko vedno izpustimo vodilne ničle in pišemo tudi tako
IO0PIN ^= 0xf0;
```