

# **Mikrokontrolleri programmeerimise spetsiifilised võtted**

Peeter Salong



# [ Sisu ]

---

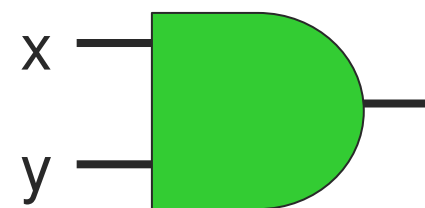
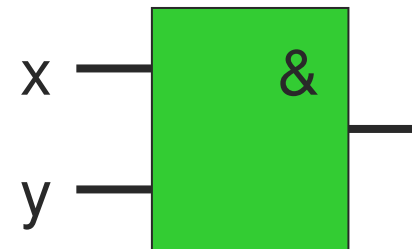
- Loogika
  - JA
  - VÕI
  - Välistav-VÕI
- Bitthaaval tehted
- 2nd süsteemi eripära



# [ JA loogika (AND) ]

x	y	x & y
0	0	0
0	1	0
1	0	0
1	1	1

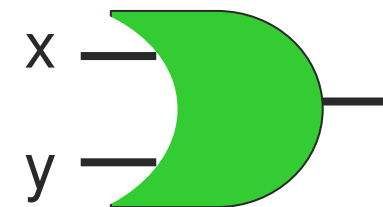
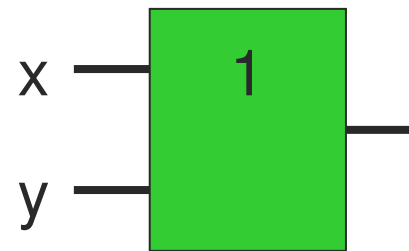
Loogikaelement



# [ VÕI loogika (OR) ]

x	y	$x \mid y$
0	0	0
0	1	1
1	0	1
1	1	1

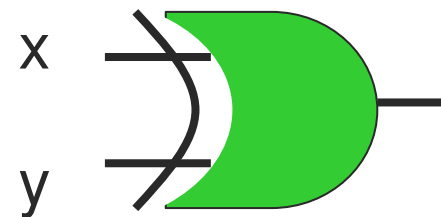
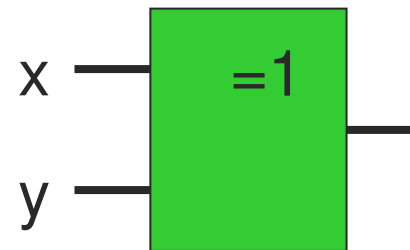
Loogikaelement



# Välistav-VÕI loogika (XOR)

x	y	$x \wedge y$
0	0	0
0	1	1
1	0	1
1	1	0

Loogikaelement



# [ Näited loogikatehetega ]

- $01100 \& 11001 = ?$
- $0xDC \mid 0x45 = ?$
- $12 \wedge 34 = ?$



# Üheks seadma (Set)

## Pordi B andmesuunaregister DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- `uint8_t test = 0x22; // 0010 0010`
- `test = test | 0x04; // 0010 0110`



# [ Nullima (Reset) ]

- `uint8_t test = 0x22; // 0010 0010`
- `test = test & ~0x20; // 0000 0010`
- `(test = test & 0xDF;)`





# [ Muutma (Toggle) ]

- `uint8_t test = 0x22; // 0010 0010`
- `test = test ^ 0x28; // 0000 1010`



# [ Nihutamised ]

- Shift left <<

$$0110 \ll 2 = 1000$$

- Shift right >>

$$0110 \gg 2 = 0001$$



# [ Näide ]

- 16-bitine arv (`test`)
- 8 bitine andmepaketi suurus

```
uint16 result = 0;
```

```
....
```

```
result = testH;
```

```
result = result << 8;
```

```
result = result | testL;
```

```
....
```



# [ 2nd süsteemi eripära ]

- 2-ga korrutamine
- **Mitte** `result = result / 2;`
- **Vaid** `result = result << 1;`
  
- 2-ga jagamine
- `result = result >> 1;`



# [ 2nd süsteemi eripära ]

- Kehtib 2 astmete korral
- $4 - \gg 2$  või  $\ll 2$
- $8 - \gg 3$  või  $\ll 3$



[Tänaan!

---

]

