

A photograph of four students in a computer lab. A man and a woman are in the foreground, smiling and looking at a computer screen. Two other students are visible behind them, also working on computers. The background shows a window with a view of a city.

Embedded Systems (NSE)

Lesweek 12 - Hoorcollege 3

Gert den Neijssel

November 2022

DE HAAGSE
HOGESCHOOL

Inhoud van deze presentatie

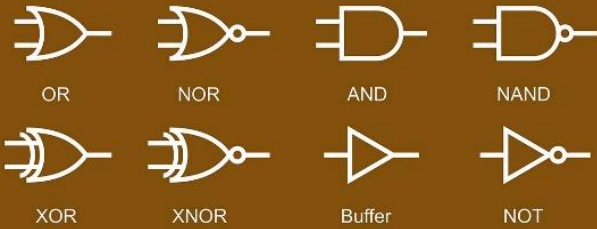
Stukje digitale techniek:

- Digitale poorten
- Circuits
- Waarheidstabellen
- Logische functies

Bewerkingen op bitniveau

- Met AND, OR en XOR

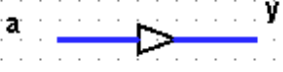
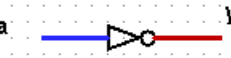
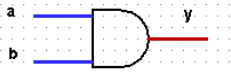
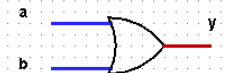

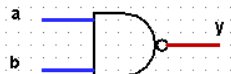
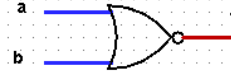

Logic Gate Symbols



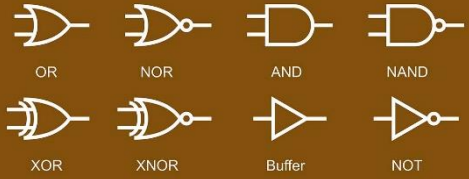
Digitale poorten

Nog meer enen (1) en nullen (0)

overzicht

buffer	$y = a$	
NOT	$y = \bar{a}$	
AND	$y = a \cdot b$	
OR	$y = a + b$	
XOR	$y = a \oplus b$	
NAND	$y = \overline{a \cdot b}$	
NOR	$y = \overline{a + b}$	
NXOR	$y = \overline{a \oplus b}$	

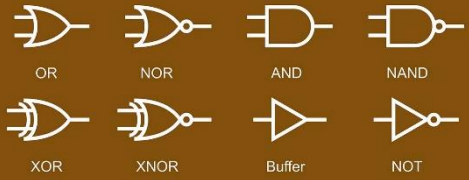
Logic Gate Symbols



Overzicht

- Bij elke poort hoort een waarheidstabel.
- Ken je bijvoorbeeld de tabel die hoort bij
- $L = \overline{A \cdot B}$?
- Herken je dan de NAND – poort hierin?

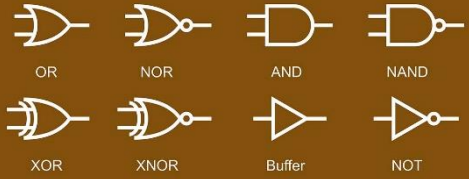
A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1



Overzicht

- Bij elke poort hoort een waarheidstabel.
- Ken je bijvoorbeeld de tabel die hoort bij
- $L = \overline{A + B}$?
- Herken je dan de NOR – poort hierin?

<i>A</i>	<i>B</i>	<i>A + B</i>
0	0	0
0	1	1
1	0	1
1	1	1



Overzicht

- Bij elke poort hoort een waarheidstabel.
- Ken je bijvoorbeeld de tabel die hoort bij
- $L = \overline{A \oplus B}$?
- Herken je dan de NXOR – poort hierin?

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Circuits (hardware)

- Een circuit is een combinatie van twee of meer poorten.
- Een circuit heeft een bepaalde functionaliteit.
- Deze functionaliteit is op verschillende manieren te ‘verwoorden’:
 - ☐ Tekstueel (requirements).
 - ☐ Waarheidstabel.
 - ☐ Logische functie.
 - ☐ Schema, tekening, simulatie.



Voorbeeld

Ontwerp een logische schakeling (= circuit) om een geluidssignaal te produceren in een auto om de bestuurder te waarschuwen bij het uitstappen dat de lichten nog branden.

Voorbeeld

Het al of niet produceren van het geluidssignaal is afhankelijk van drie sensoren:

- A , onder de bestuurdersstoel meet het al (1) of niet (0) bezet zijn van de stoel.
- B , meet het open (1) of gesloten (0) zijn van de deur aan de kant van de bestuurder.
- C , meet wel (1) of geen (0) sleutel in het startcontact.

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Voorbeeld

Requirements:

- Het geluidssignaal dient gegeven te worden ($X = 1$), zodra de bestuurder de sleutel uit het startcontact haalt, ook tijdens het openen van de deur, zelfs wanneer de sleutel nog in het startcontact zit.
- Er is geen geluidssignaal ($X = 0$) wanneer de deur dicht is en er geen bestuurder in het voertuig zit.

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Waarheidstabel

A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Waarheidstabel

Er is **geen** geluidssignaal ($X = 0$) wanneer de **deur dicht** is en er geen bestuurder in het voertuig zit.

Deur dicht: $B = 0$ en Geen bestuurder: $A = 0$

A	B	C	X
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Waarheidstabel

Het geluidssignaal dient gegeven te worden ($X = 1$), zodra de bestuurder de **sleutel** uit het startcontact haalt.

Geen sleutel: $C = 0$.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	
1	0	0	1
1	0	1	
1	1	0	1
1	1	1	

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Waarheidstabel

Het geluidssignaal dient gegeven te worden ($X = 1$), ook tijdens het **openen van de deur**, zelfs wanneer de **sleutel nog in het startcontact zit**.

Deur open: $B = 1$ en wel sleutel: $C = 1$.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	
1	0	0	1
1	0	1	
1	1	0	1
1	1	1	

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	
1	1	0	1
1	1	1	1

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Waarheidstabel

Eén situatie is nog niet benoemd: $A = 1, B = 0, C = 1$.
Wat is dan de situatie? En wil je dan geluid of niet?

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	
1	1	0	1
1	1	1	1

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

vervolg?

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- In de volgende fase proberen we vanuit de waarheidstabel een logische functie te ontwerpen.
- De aanpak hiervoor is te kijken naar de enen in de antwoordkolom, hier X .

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- De aanpak hiervoor is te kijken naar de enen in de antwoordkolom, hier X .
- Bij elke $X = 1$, hoort een bepaalde (vaste combinatie) van waarden van A , B en C .
- Zo is bijvoorbeeld $X = 1$ als de combinatie $A = 0$ en $B = 1$ en $C = 0$ geldt.

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- Zo is bijvoorbeeld $X = 1$ als de combinatie $A = 0$ en $B = 1$ en $C = 0$ geldt.
- Hoe maak je hier een functie van?
Een logische **AND** is in formulevorm de logische vermenigvuldiging:
- $X = A \cdot B \cdot C$, maar met de bovenstaande combinatie voor de inputs wordt dit:
 $X = 0 \cdot 1 \cdot 0 = 0 \quad ?$

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- $X = A \cdot B \cdot C$, maar met de gegeven combinatie voor de inputs wordt dit $X = 0 \cdot 1 \cdot 0 = 0$
- Hoe kunnen we uitkomen op $X = 1$?
- Alleen als alle ingangen 'hoog' zijn.
- Dus als: $X = \bar{A} \cdot B \cdot \bar{C}$
- Of wel: $X = \bar{0} \cdot 1 \cdot \bar{0} = 1 \cdot 1 \cdot 1 = 1$
- Dit is wel het gewenste resultaat.

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- De volgende combinatie welke $X = 1$ oplevert, is $A = 0, B = 1, C = 1$.
- Overeenkomstig moet dit dan worden vertaalt als $X = \bar{A} \cdot B \cdot C$

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- Zo zijn er nog eens drie combinaties welke $X = 1$ opleveren:
- $X = A \cdot \bar{B} \cdot \bar{C}$
- $X = A \cdot B \cdot \bar{C}$
- $X = A \cdot B \cdot C$
- Hoe worden deze vijf mogelijkheden voor $X = 1$ met elkaar gecombineerd?

logische functie

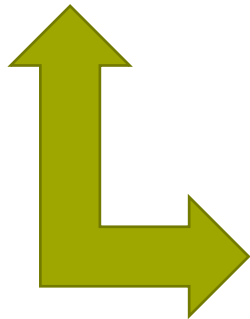
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- Hoe worden deze vijf mogelijkheden voor $X = 1$ met elkaar gecombineerd?
- In elk geval niet met een logische AND, want op een zeker moment kan het niet zo zijn dat er twee (of meer) rijen in de waarheidstabel gelijktijdig van toepassing zijn!
- De vijf mogelijkheden worden derhalve met elkaar gecombineerd met een **logische OR**.

logische functie

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- De vijf mogelijkheden worden derhalve met elkaar gecombineerd met een logische OR.



$$X = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$



Voorbeeld

Ontwerp een logische schakeling (=circuit) om een geluidssignaal te produceren in een auto om de bestuurder te waarschuwen bij het uitstappen dat de lichten nog branden.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

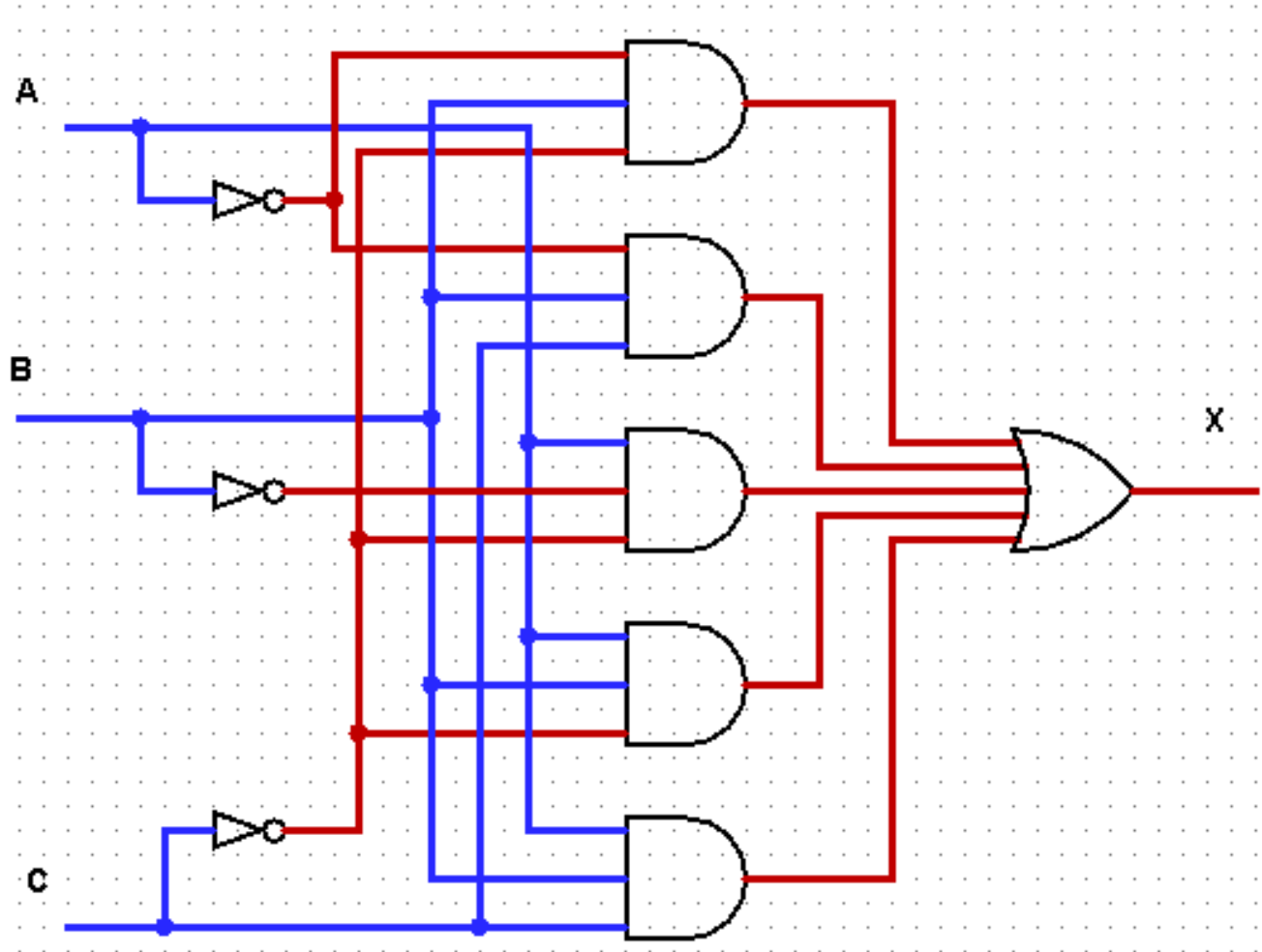
$$X = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

Circuit?



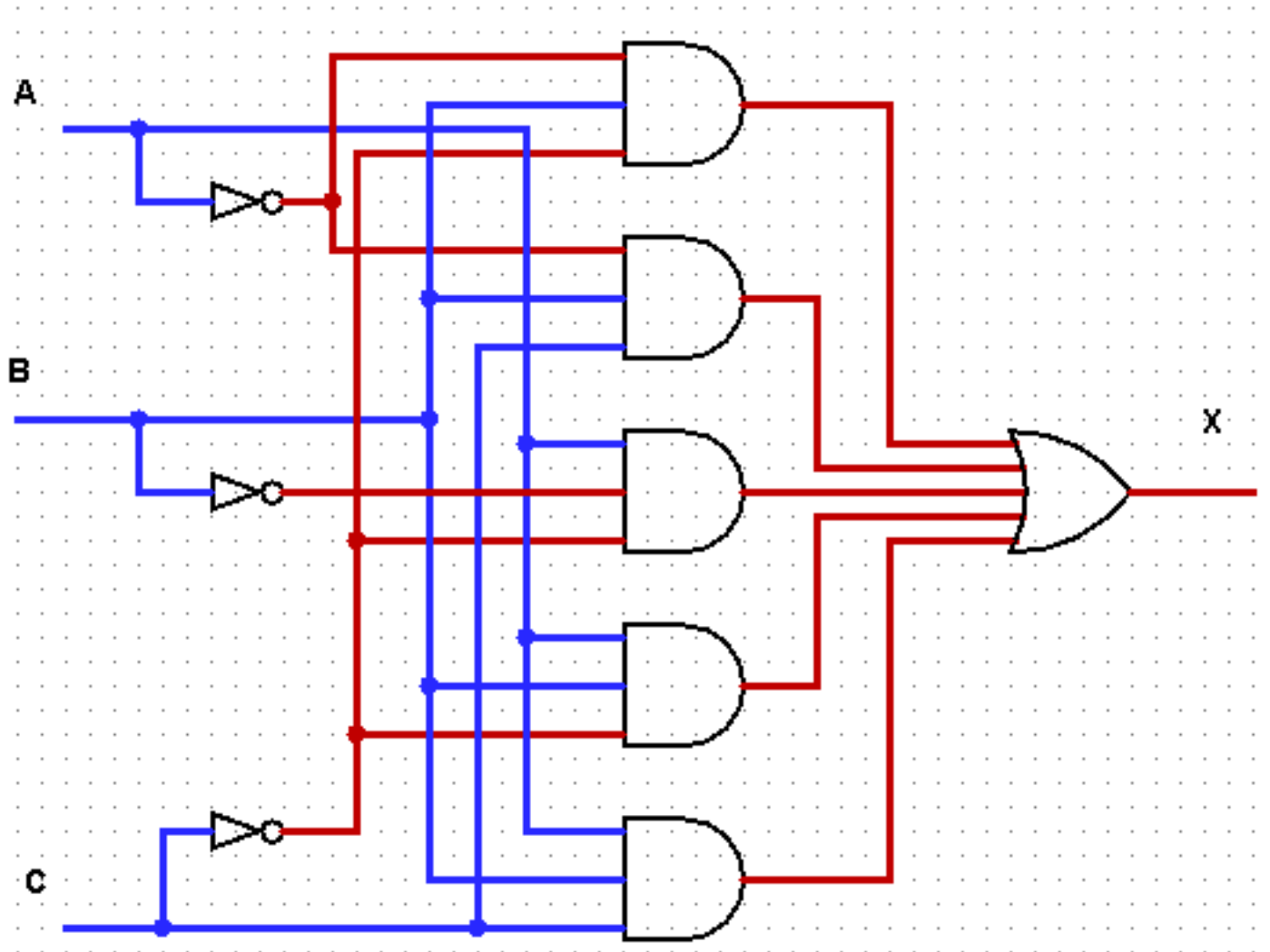
Circuit

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC\bar{C} + ABC$$





Circuit eigenschappen



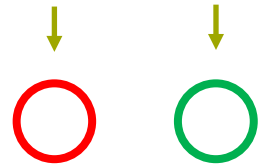
#poorten: ?
#ingangen: ?

#poorten: 9
#ingangen: 23

Circuit vereenvoudigen

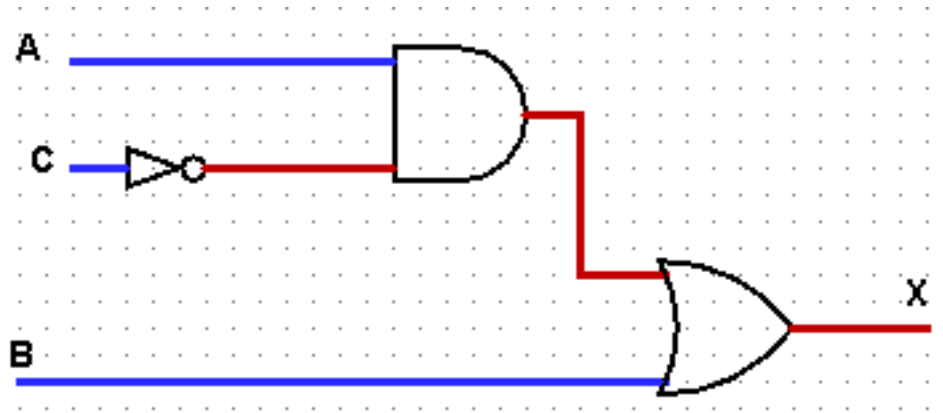
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

- Bij vier combinaties geldt, als $B = 1$, dan $X = 1$ ongeacht de waarde van A en/of C .
- Voor de 1-na laatste 1 in kolom X , geldt ook nog iets bijzonders, namelijk als $A = 1, C = 0$ doet de waarde van B er niet toe.
- Voor de vereenvoudigde functie X' geldt dan ook:
- $X' = B + A\bar{C}$.



Circuit vereenvoudigen

$$X' = B + A\bar{C}.$$



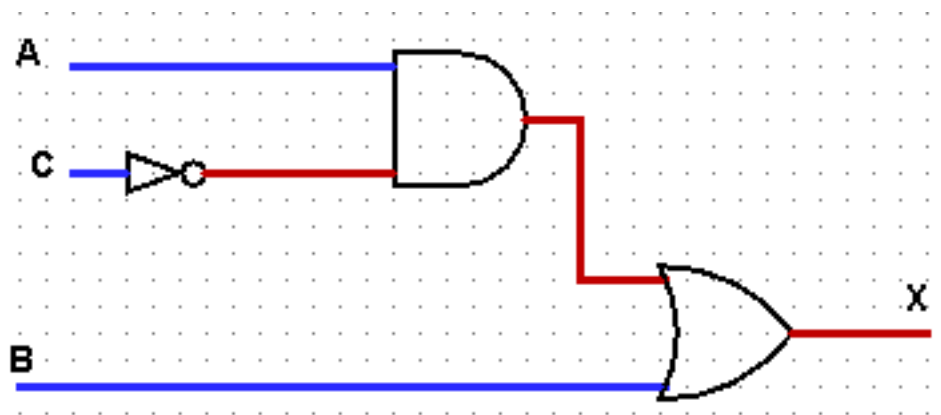
#poorten: ?
#ingangen: ?

#poorten: 3
#ingangen: 5

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Circuit vereenvoudigen

$$X' = B + A\bar{C}.$$



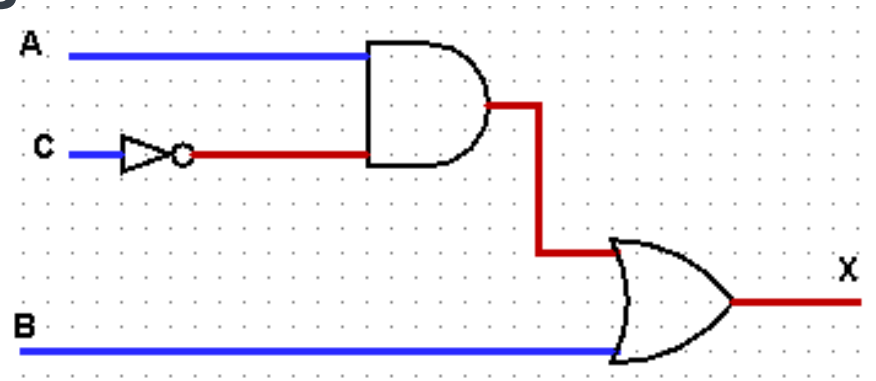
Wat is nu de functionele betekenis van X' ?

Wanneer, als de autolampen aanstaan, komt er een geluidsignaal?

1. De deur gaat open.
2. De stoel is bezet en er is geen sleutel in het startcontact.

Stoel (A): 1 = bezet
Deur (B): 1 = open
Contact (C): 1 = wel sleutel

Hoe kan dit in een programmeeromgeving gedaan worden?



```
#define BEZET true
#define LEEG false
#define OPEN true
#define DICHT false
#define SLEUTEL_IN true
#define SLEUTEL_UIT false
#define AlarmPin 0
```

$$X' = B + A\bar{C}.$$

```
boolean stoel = BEZET;
boolean deur = OPEN;
boolean contact = SLEUTEL_IN;
```

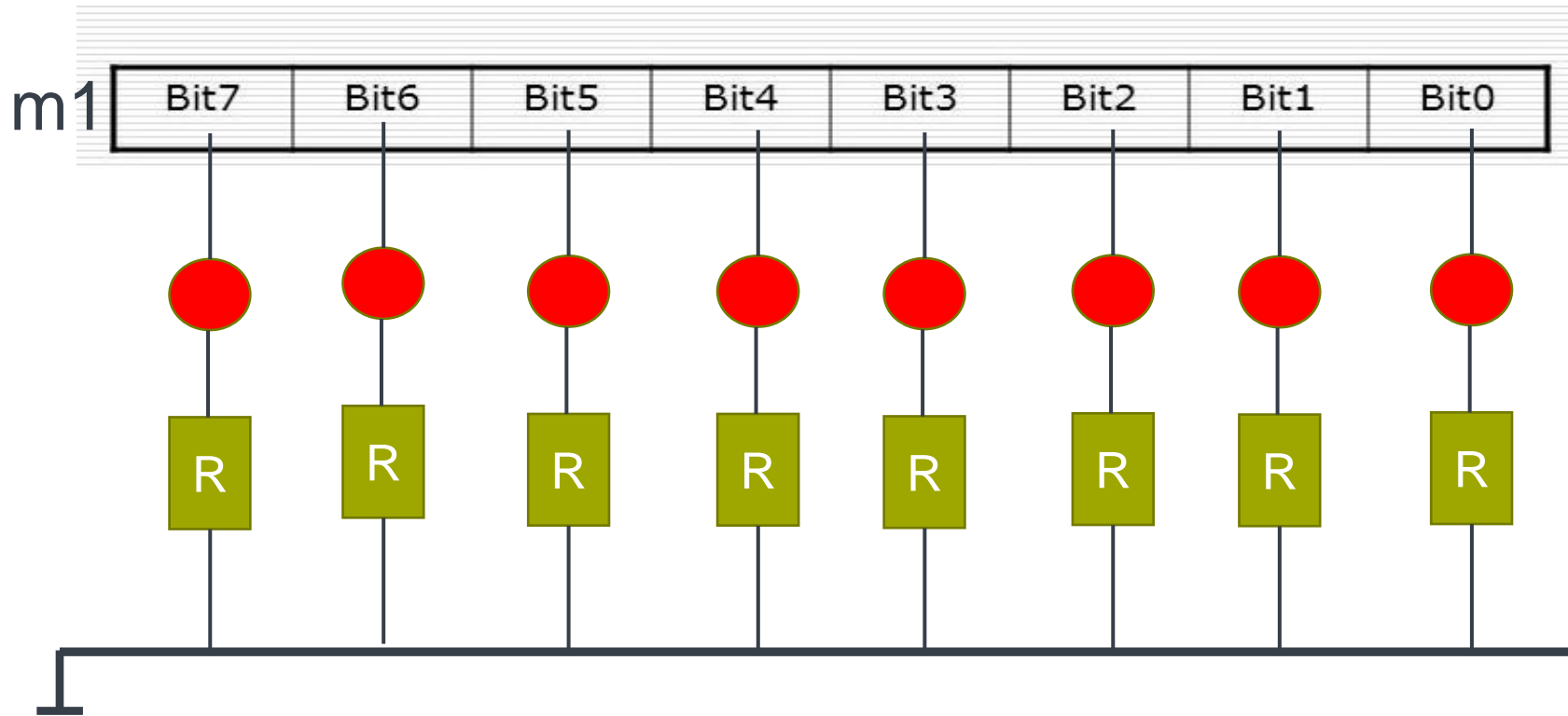
```
void loop() {
    if( deur || ( stoel && ! contact ) )
        digitalWrite(AlarmPin,HIGH);
    else
        digitalWrite(AlarmPin,LOW);
}
```

Bewerkingen doen op bits

m.b.v OR, AND en XOR bewerkingen

8 LEDs worden aangesloten

- We willen de LEDs:



- aan- en uitzetten.
- controleren is aan.
- toggelen.

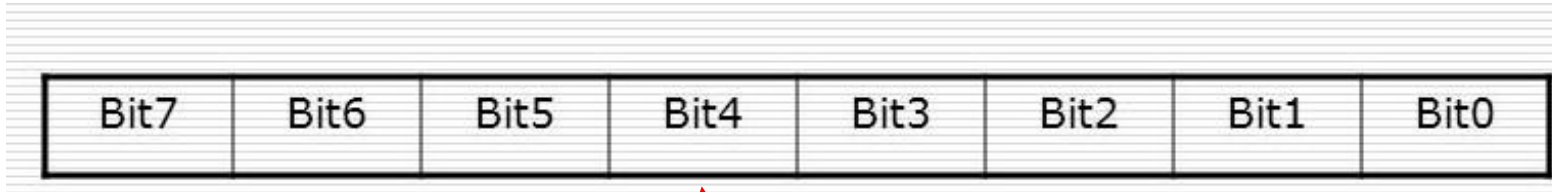
Bewerkingen doen op de bits

samenvatting

bewerking	bitwise-operator	masker	opmerking
check bit	AND	00..010..0	Is het resultaat 0, dan is het oorspronkelijk bit een 0; is het resultaat niet-0, dan is het bit een 1.

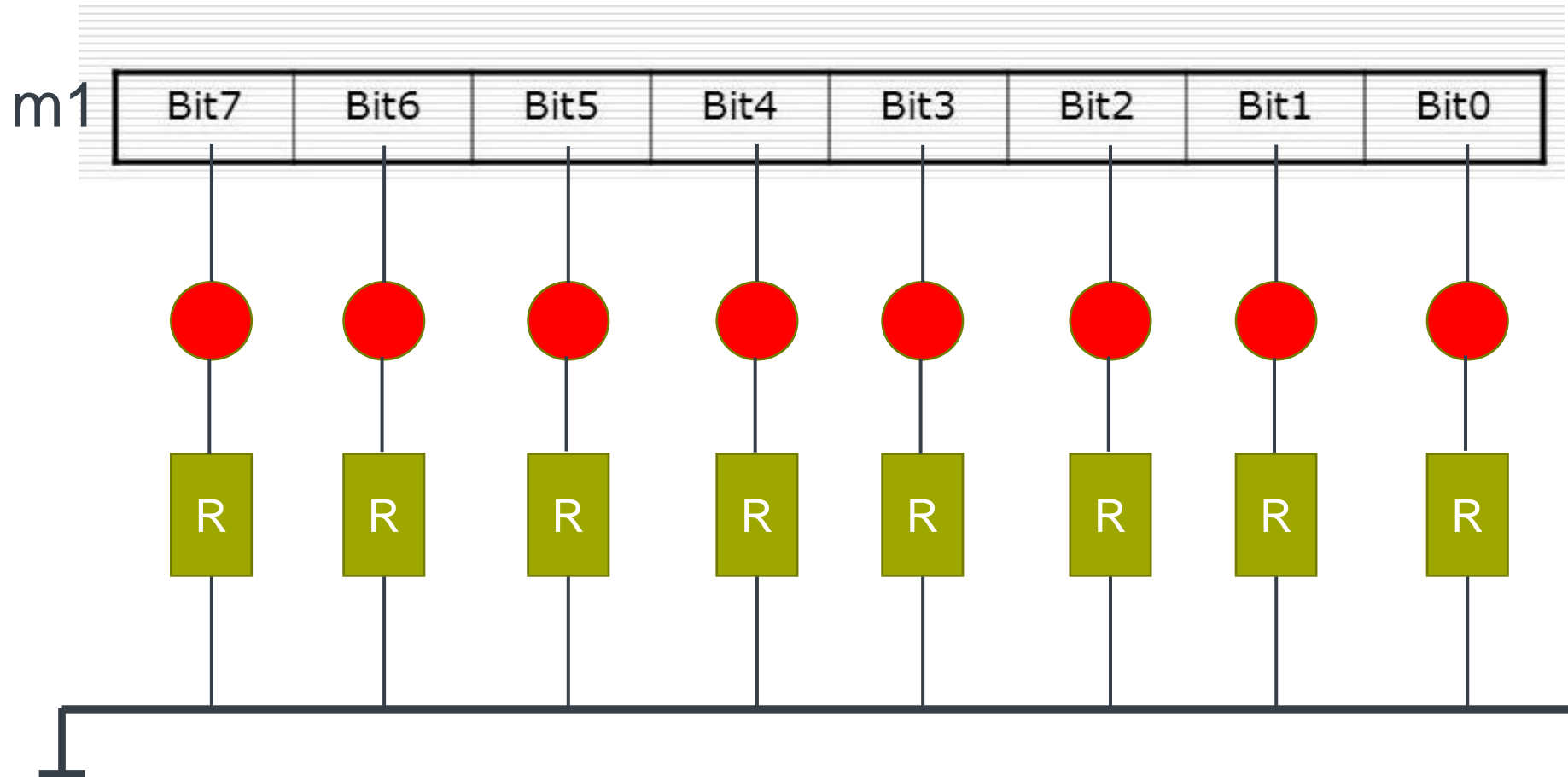
vooraf

- De bitnummering in 1 byte is als volgt:



- Wanneer we in het vervolg praten over het 5^e bit (van rechts) dan bedoelen we Bit4.
- Informatie nuttig voor een willekeurige microcontroller, wordt opgeslagen in **specifieke stukjes geheugen**, register genoemd.
- Elk **register** is te benaderen op basis van zijn **unieke**, specifieke **naam**.
- De meeste registers in een willekeurige (kleine) microcontroller zijn 1 byte groot.

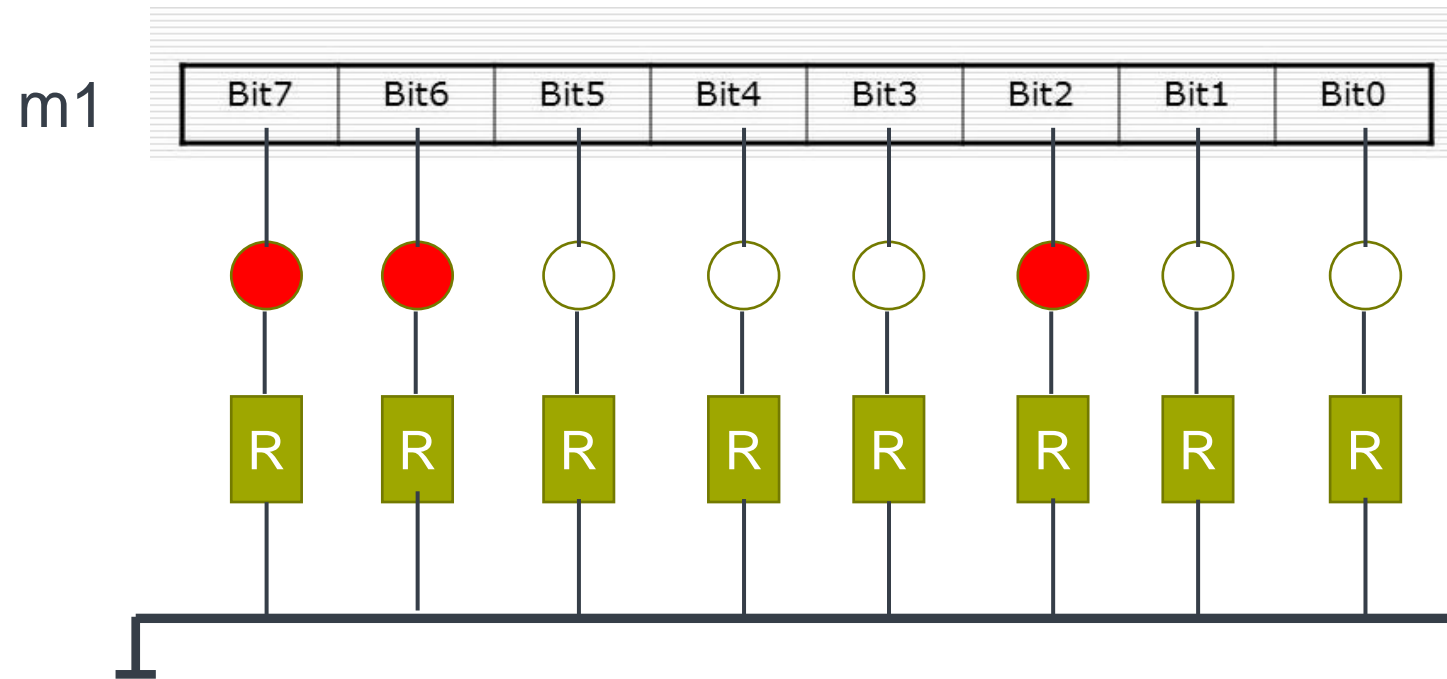
8 LEDs worden aangesloten



- We willen de LEDs aan en uitzetten, dit gebeurt door het overeenkomstige bit te *setten* of te *resetten*.

Toekennen van een waarde

- Laten we voor de LEDs op bit2, bit 6 en bit 7 aanzetten.



- `m1 = 0b11000100;`
of
- `m1 = 0xC4;`
of
- `m1 = 196;`

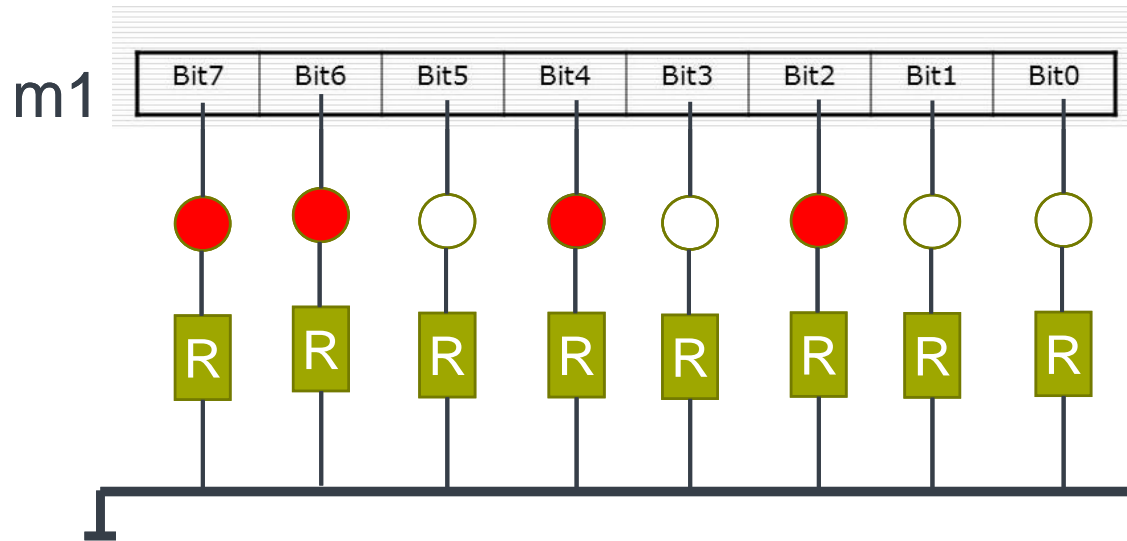
- `m1` moet de waarde 11000100 krijgen.
- Hoe komt deze waarde in `m1`?

Het aanzetten van een bit

(Zonder de andere bits te beïnvloeden).

<i>A</i>	<i>B</i>	<i>A B</i>
0	0	0
0	1	1
1	0	1
1	1	1

- Laten we voor de LED4 aanzetten.



m1 = 0b11000100; **A**

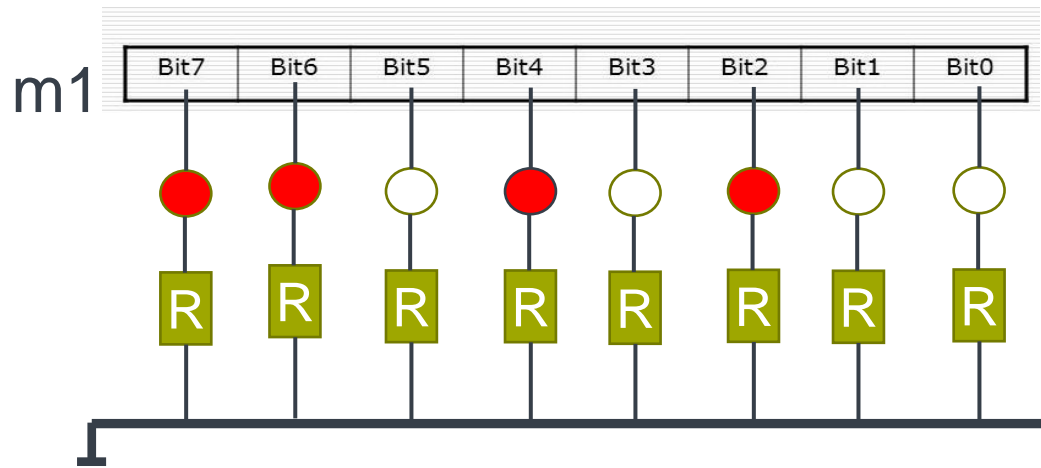
M1_{moet} worden 0b11010100;

- m1 moet i.p.v van 11000100 de waarde 11010100 krijgen.
- Hoe krijgen we dit voor elkaar?

Het aanzetten van een bit

(Zonder de andere bits te beïnvloeden).

- Het aanzetten van LED4.



<i>A</i>	<i>B</i>	<i>A B</i>
0	0	0
0	1	1
1	0	1
1	1	1

m1 = 0b1 1 0 ~~0~~ 0 1 0 0; (huidige situatie)

0b0 0 0 **1** 0 0 0 0 | (**of** bewerking)

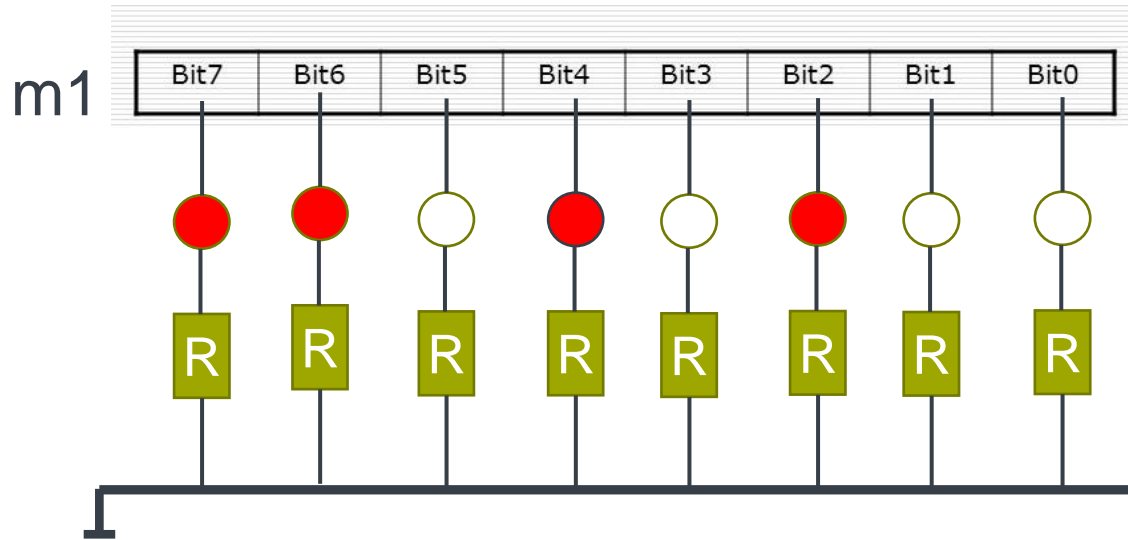
M1_{moet} worden 0b1 1 0 **1** 0 1 0 0; (gewenste situatie)

Het aanzetten van een bit

(Zonder de andere bits te beïnvloeden).

<i>A</i>	<i>B</i>	<i>A B</i>
0	0	0
0	1	1
1	0	1
1	1	1

- Laten we voor de LED4 aanzetten.



m1 = 0b11000100;

0b00010000; |

M1_{moet} worden 0b11010100;

m1 = m1 | 0b00010000; => m1 |= 0b00010000;

Of

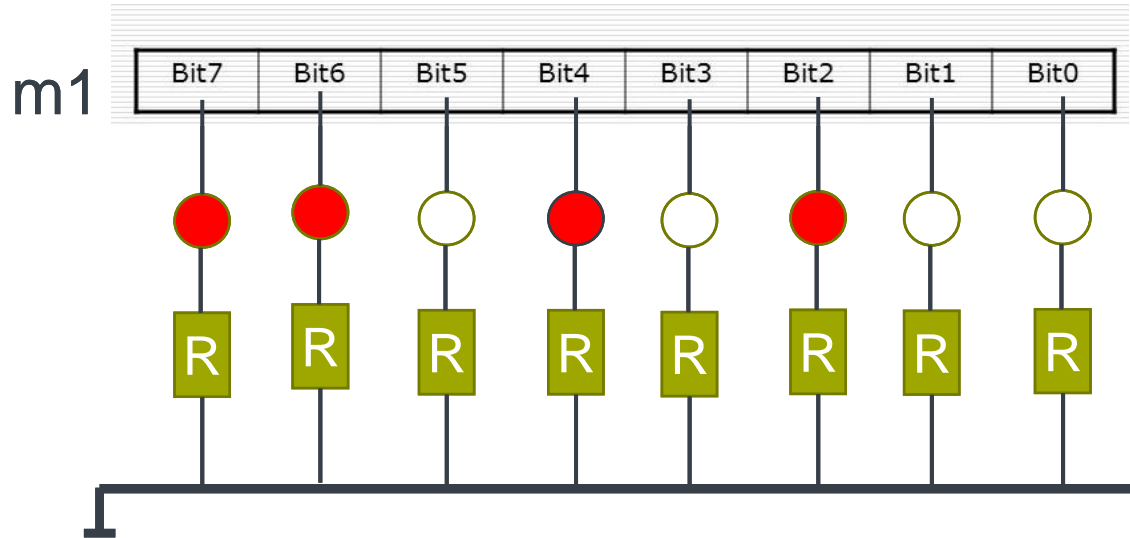
m1 = m1 | 0x10; => m1 |= 0x10;

Het uitzetten van een bit

(Zonder de andere bits te beïnvloeden).

<i>A</i>	<i>B</i>	<i>A & B</i>
0	0	0
0	1	0
1	0	0
1	1	1

- Laten we voor de LED6 uitzetten.



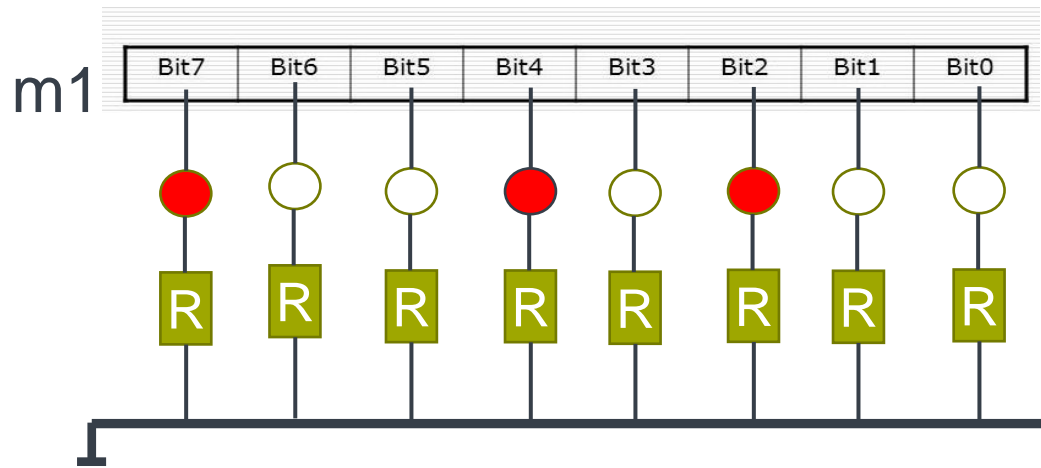
m1 = 0b1**1**000100; **A**
M1_{moet} worden 0b1**0**010100;

- m1 moet i.p.v van 1**1**010100 de waarde 1**0**010100 krijgen.
- Hoe krijgen we dit voor elkaar?

Het uitzetten van een bit

(Zonder de andere bits te beïnvloeden).

- Het uitzetten van LED6.



<i>A</i>	<i>B</i>	<i>A & B</i>
0	0	0
0	1	0
1	0	0
1	1	1

m1 = 0b1 **1** 0 1 0 1 0 0; (huidige situatie)

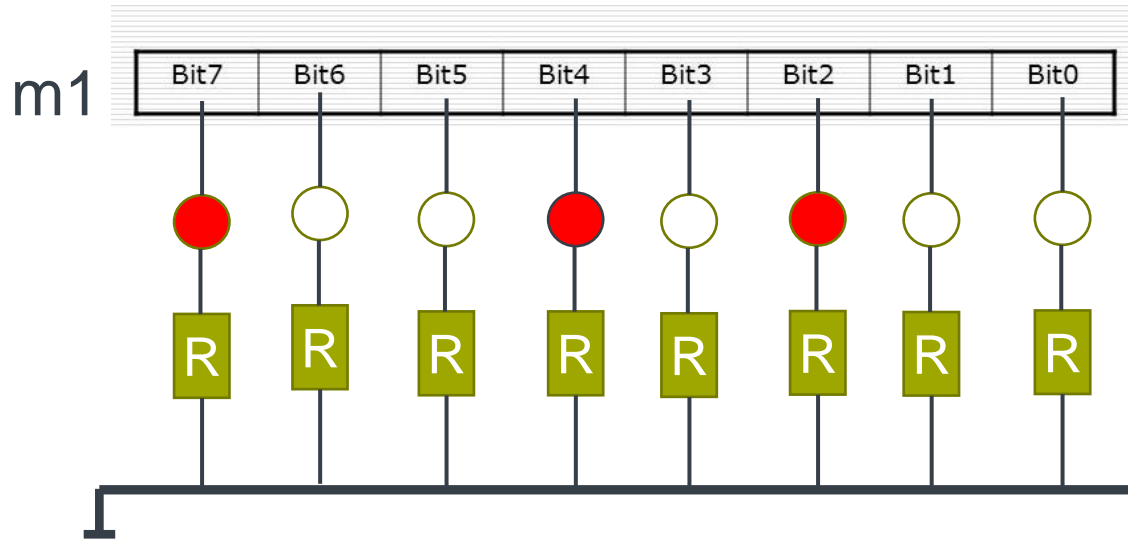
0b1 **0** 1 1 1 1 1 1 **&** (**AND** bewerking)

M1 moet worden 0b1 **0** 0 1 0 1 0 0; (gewenste situatie)

Het uitzetten van een bit

(Zonder de andere bits te beïnvloeden).

Het uitzetten van LED6.



<i>A</i>	<i>B</i>	<i>A & B</i>
0	0	0
0	1	0
1	0	0
1	1	1

`m1 = 0b11010100;`
`0b10111111; &`
`M1moet worden 0b10010100;`

`m1 = m1 & 0b1011.1111; => m1 &= 0b1011.1111;`

Of

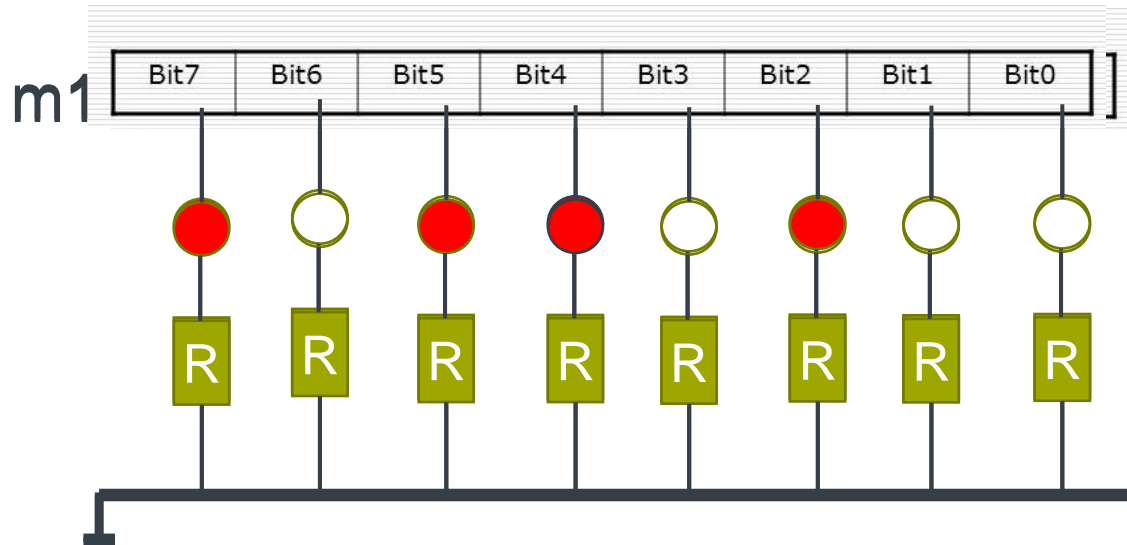
`m1 = m1 & 0xBF; => m1 &= 0xBF;`

Het wisselen van een bitwaarde

(Zonder de andere bits te beïnvloeden).

<i>A</i>	<i>B</i>	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0

Laten we LED5 aan- en LED2 uitzetten



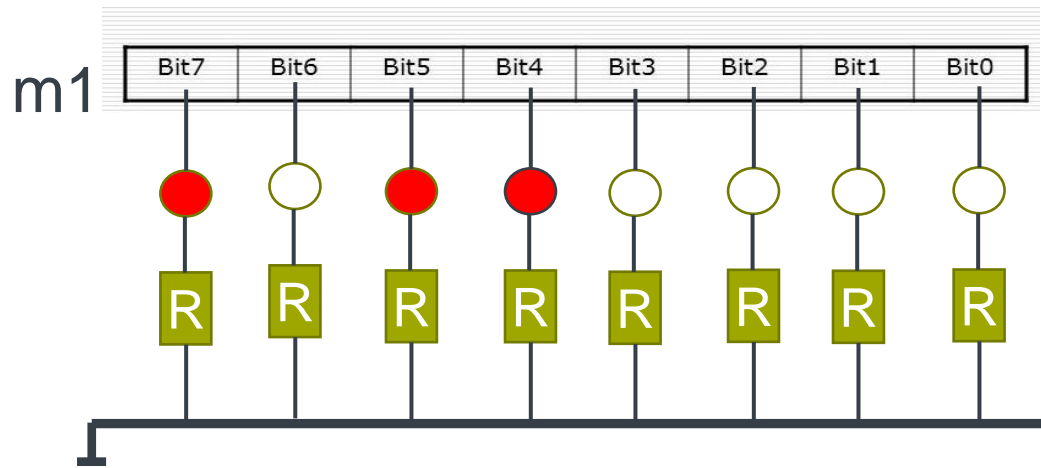
`m1 = 0b11000100; A`
`m1` moet worden `0b10110000;`

- `m1` moet i.p.v. van `10010100` de waarde `10110000` krijgen.
- Hoe krijgen we dit voor elkaar?

Het wisselen van een bitwaarde

(Zonder de andere bits te beïnvloeden).

LED2 en LED5 krijgen een andere waarde.



<i>A</i>	<i>B</i>	<i>A</i> ^ <i>B</i>
0	0	0
0	1	1
1	0	1
1	1	0

m1 = 0b1 0 0 1 0 1 0 0; (huidige situatie)

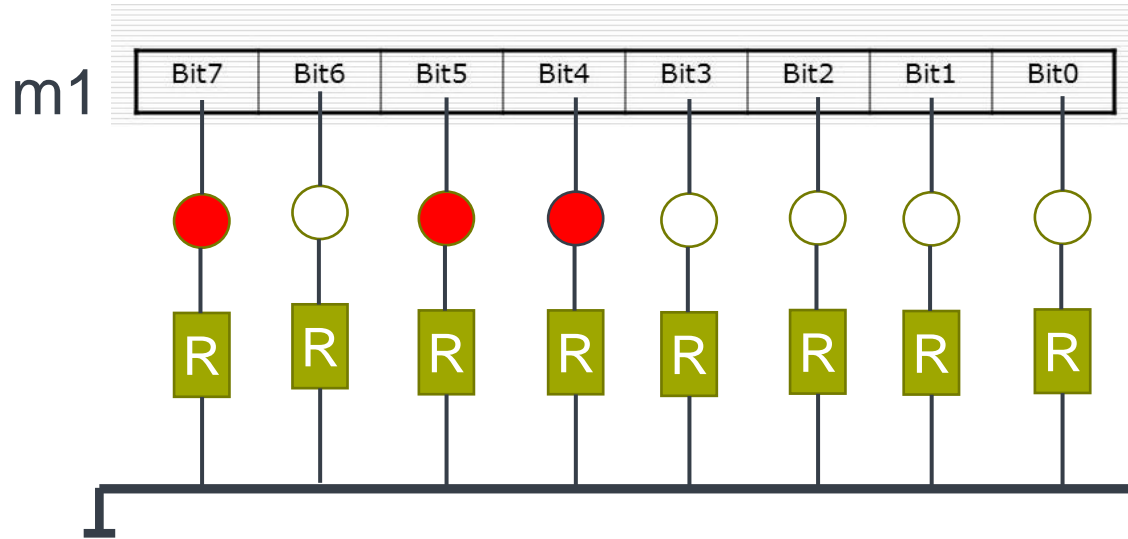
0b0 0 1 0 0 1 0 0 ^ (toggle bewerking)

M1 moet worden 0b1 0 1 1 0 0 0 0; (gewenste situatie)

Het wisselen van een bitwaarde

(Zonder de andere bits te beïnvloeden).

LED2 en LED5 krijgen een andere waarde.



<i>A</i>	<i>B</i>	<i>A</i> ^ <i>B</i>
0	0	0
0	1	1
1	0	1
1	1	0

`m1 = 0b11010100;`
`0b10111111;` ^
`M1` moet worden `0b10010100;`

`m1 = m1 ^ 0b0010.0100; => m1 ^= 0b0010.0100;`

Of

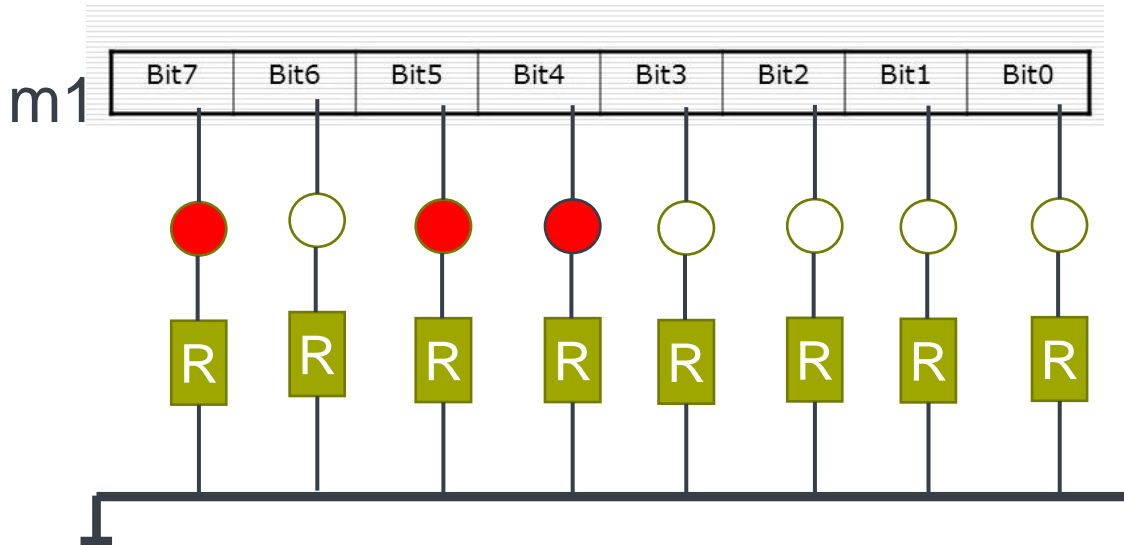
`m1 = m1 ^ 0x24; => m1 ^= 0x24;`

Het controleren van een bitwaarde

(Heef een bit de waarde 0 of 1).

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Stel we willen weten of LED3 aan of uit is.



is(van m1(0b10110000) bit 3 een 1)

ja:....

ander:...

Masker

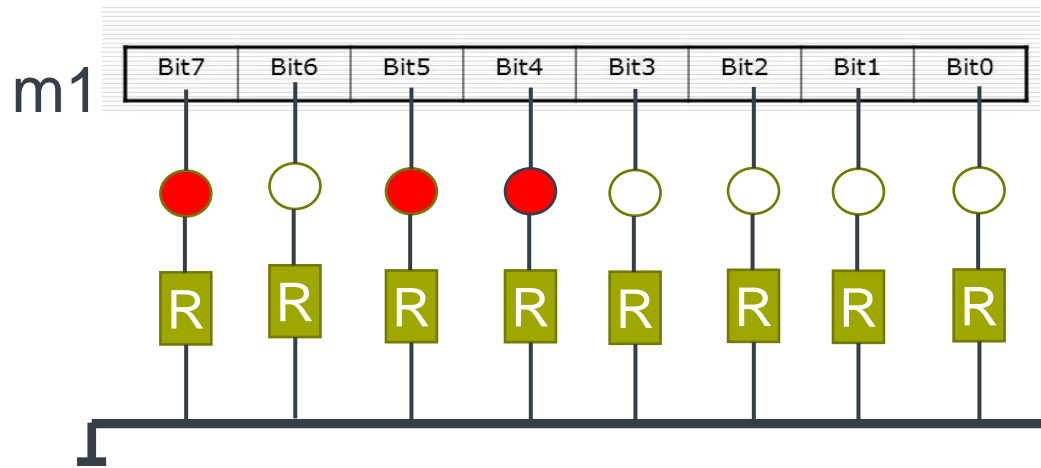
- Is bit 3 van m1 (10110000) nu een 0 of een 1.
- Hoe krijgen we dit voor elkaar?

Het controleren van een bitwaarde

(Heef een bit de waarde 0 of 1).

<i>A</i>	<i>B</i>	<i>A & B</i>
0	0	0
0	1	0
1	0	0
1	1	1

Stel we willen weten of LED3 aan of uit is.



`m1 = 0b1 0 0 1 0 1 0 0; (huidige situatie)`

`0b0 0 0 0 1 0 0 0 & (AND bewerking)`

`0 0 0 0 0 0 0 0`

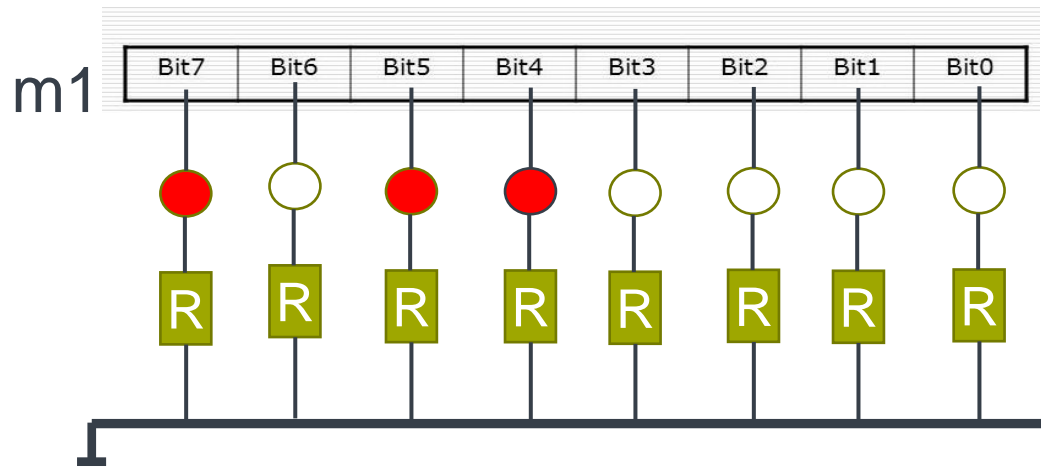
Uitkomst is **waar** of **niet waar**

Het controleren van een bitwaarde

(Heef een bit de waarde 0 of 1).

<i>A</i>	<i>B</i>	<i>A & B</i>
0	0	0
0	1	0
1	0	0
1	1	1

Stel we willen weten of LED3 aan of uit is.



m1 = 0b1 0 0 1 **1** 1 0 0 (huidige situatie)

0b0 0 0 0 **1** 0 0 0 **&** (**AND** bewerking)

0 0 0 0 **1** 0 0 0

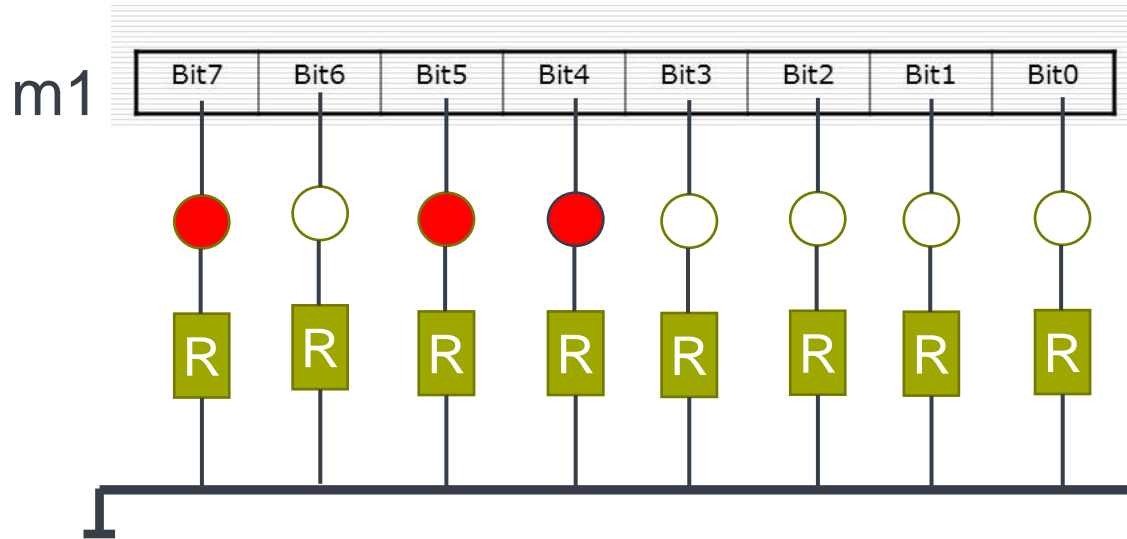
Uitkomst is **waar** of **niet waar**

Het controleren van een bitwaarde

(Heef een bit de waarde 0 of 1).

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Stel we willen weten of LED3 aan of uit is.



```
m1 = 0b11010100;
      0b00001000; &
false 00000000;
```

```
if( m1 & 0b00001000 ) {
    LED is aan
}
else {
    LED is uit
}
```

Masker

```
if( m1 & 0x08 ) {
    LED is aan
}
else {
    LED is uit
}
```



A grayscale photograph of a group of students in a library or study hall. In the foreground, a young woman with long dark hair, wearing a denim jacket, is leaning over a table, pointing at a document. Two young men are sitting at the table, looking at the document. Another young woman is sitting next to them, looking at a laptop. In the background, other students are visible, some sitting at tables and others standing. The text 'DE HAAGSE HOGESCHOOL' is overlaid in the center of the image.

DE HAAGSE HOGESCHOOL

let's change
YOU. US. THE WORLD.