

# Docenten aan het woord

Kevin Krul

# Wie ben ik?

- Kevin Krul
  - 32 jaar
  - Rotterdam
- 
- Dr. Mollercollege, Waalwijk
  - Informatica lessen bovenbouw

Heeft die echt zijn complete presentatie in zwart/wit?

JA!

# Mijn geschiedenis

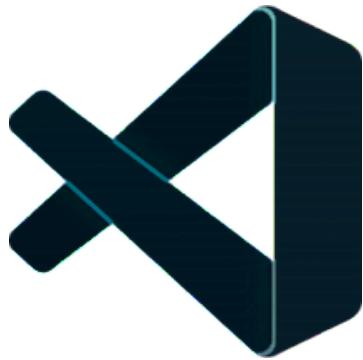
- Een jaar back-end PHP ontwikkelaar
- Twee jaar full-stack Java ontwikkelaar
- Nu nog steeds, freelance full-stack web development

*Java, PHP, Javascript, NodeJS, React, Angular, SQL, MongoDB, Kotlin, React Native*

# Ik als docent

Vanuit mijn persoonlijke geschiedenis een sterke focus op programmeren.

Met de nieuwste tools en de nieuwste functionaliteit.



```
:
$uitkomst = queryUpdate ("UPDATE gebruiker SET mail :  

if ($uitkomst == true) {  

echo "<br> E-mail geupdate";  

:  

:  

} else {  

echo "E-mail updaten mislukt";  

}  

:  

:
```

# Hoe doe ik dat?

- Praktische onderwerpen schrijf ik zelf uitleg en opdrachten voor
- Ik doe zoveel mogelijk onderwerpen direct in het programmeren verwerken
- Ik probeer mijn programmeerlessen zo modern mogelijk te houden

# HTML?

Iedere web-ontwikkelaar zal je vertellen; mobile first!

Dit sluit ook beter aan bij de belevingswereld van de leerling.  
En je hebt gelijk de nieuwste CSS technologieën nodig; flexbox, grid, etc..

Fundament 2019 -  
Kernprogramma > B: Grondslagen >  
Algoritmen (B1) > 1. Algoritmen

Toevoegen als favoriet

## 1.1. Inleiding

Algoritmen zijn overal

0. Voor de  
docent

1.  
Algoritmen

1.1.  
Inleiding

1.2.  
Kaarten  
sorteren

1.3.  
Sorteeralgor

1.4.  
Schematisch  
weergave  
van een  
algoritme

1.5. Hoe  
goed is  
een  
algoritme?

Als je in huis iets kwijt  
bent, bijvoorbeeld je  
portemonnee, dan ga je  
op zoek. Je probeert je te  
herinneren waar je de  
portemonnee allemaal  
neergelegd kan hebben.  
Dan ga je kijken of hij daar  
ligt. Kun je je  
portemonnee op die  
manier niet vinden? Dan  
moet je een manier  
bedenken waarop je zo  
efficiënt mogelijk het hele  
huis kunt doorzoeken.





# B1: Grondslagen, algoritmes

Als leerlingen weten wat een array is en hoe een FOR loop werkt.

Als eerste dus veel instructie, los van algoritmes

Dan laat ik ze eerst zelf een sorteeralgoritme bedenken in hun hoofd en uitschetsen op papier en daarna programmeren.

Daarna bespreken we hun bedachte oplossingen en hebben we het over dingen als efficiëntie vanuit de methode.

```
int[] sortme = new int[] { 1, 32, 5, 55, 2, 44, 123, 78 };

for (int i = 0; i < sortme.Length; i++)
{
    for (int x = 0; x < (sortme.Length -1); x++)
    {
        if (sortme[x] > sortme[x + 1])
        {
            int templeft = sortme[x];
            int tempright = sortme[x + 1];
            sortme[x] = tempright;
            sortme[x + 1] = templeft;
        }
    }
}

foreach (int i in sortme)
{
    MessageBox.Show(i.ToString());
}
```

Ik zeg natuurlijk niet dat dit de beste methode is voor iedereen.  
Het heeft wel wat nadelen.

- Lastiger voor leerlingen
- Vergt veel zelfstandigheid
- Kost meer tijd

Maar ik vind het fijn...

- Leerlingen werken, docent minder
- Ervaring opdoen met programmeren
- Sneller differentiëren

*Ja, uiteindelijk laat ik ze wel zien dat Array.Sort() ook bestaat ;)*

# Theorie lessen

- Mijn theorie lessen zijn een stuk vlotter
- Onderwerpen die niks met programmeren te maken hebben blijven over

# Theorie lessen

Twee opties:

- De leerlingen werken geheel zelfstandig zonder klassikale uitleg C1-2
- De leerlingen maken opdrachten na korte klassikale uitleg C4

Een afbeelding bestaat uit pixels met een kleur.  
Zoals in het vorige hoofdstuk besproken.

Afbeeldingen die op deze manier worden opgeslagen  
noemen wij bitmap afbeeldingen.

Voorbeeld:

.bmp

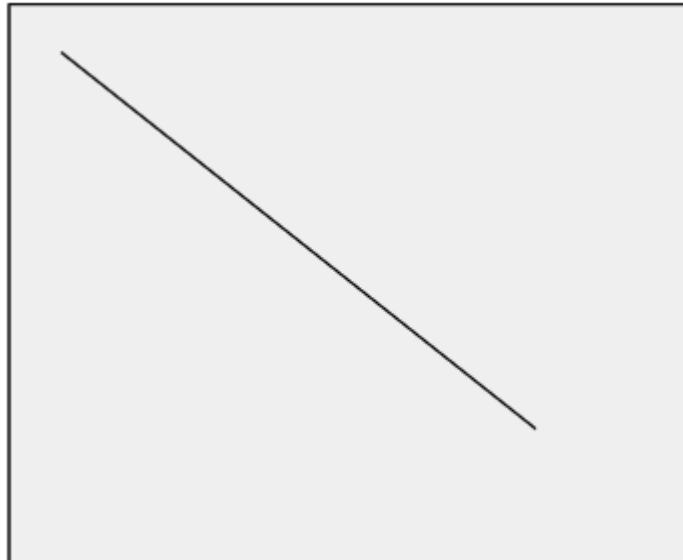
.jpg

.png

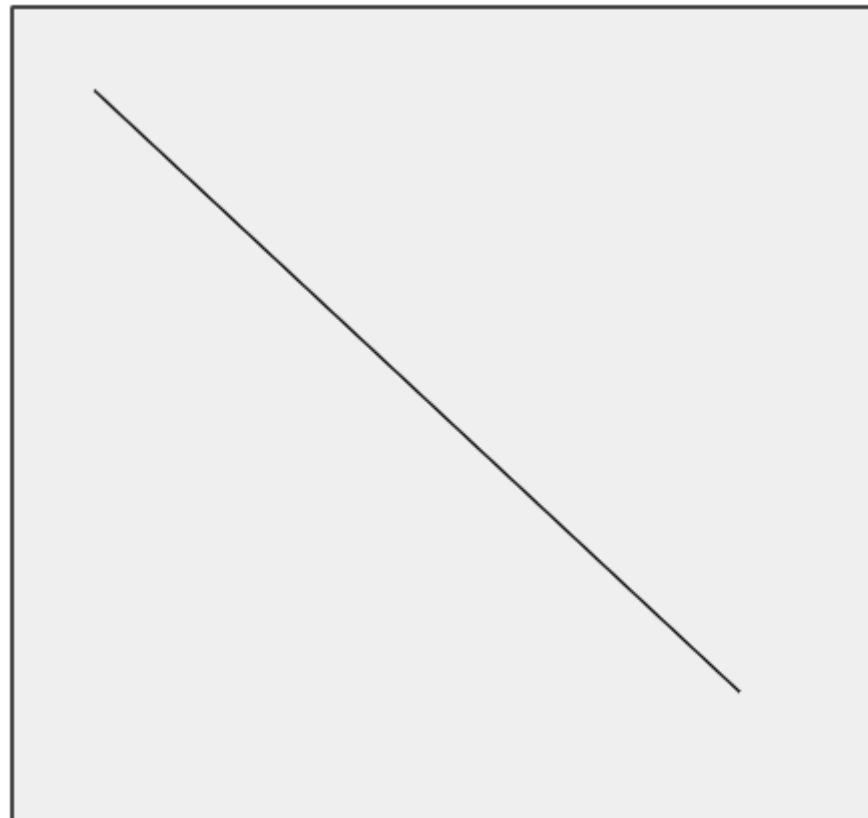
etc...

Een andere optie is een vector bestand.  
Hier wordt de afbeelding niet opgebouwd door pixels  
Maar door een formule die te schalen is.

Bijvoorbeeld, maak een lijn van 10% links schuin naar beneden tot 30% rechts.



Omdat dit niet per pixel is kan het makkelijker worden geschaald. Als je de afbeelding groter maakt blijft het beeld scherp doordat de formule opnieuw berekend wordt.



Een vector is alleen zinvol voor een afbeelding met simpele lijnen en vormen. Een foto heeft te veel lijnen en te kleine details om dit goed te laten berekenen.  
Voor een logo is dit wel handig.



Maak de bijbehorende vragen op Fundament-online.nl

C4

Hoofdstuk 3: Beeld en geluid

# Mijn PTA

Havo 4:

Examen	Domeinen	Weging
PO HTML	A, F, O	20
SE Informatie Theorie	C, F	10
SE Architectuur Theorie	A, E	10
PO Programmeren	A, D, B	10

Havo 5:

SE Database & SQL	A, C, H	15
PO PHP & SQL	A, D, H	15
SE Netwerken	E, L	20

OVER MIJ!  
HOBBY'S  
MUZIEK  
INFORMATICA?

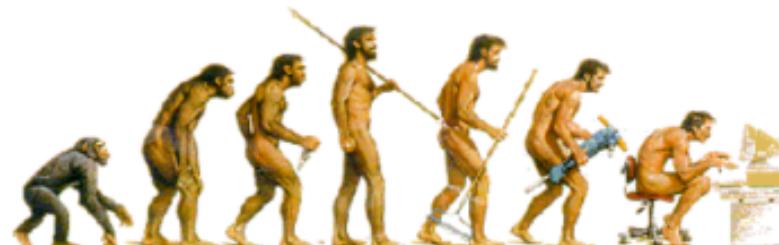
# DANIQUE

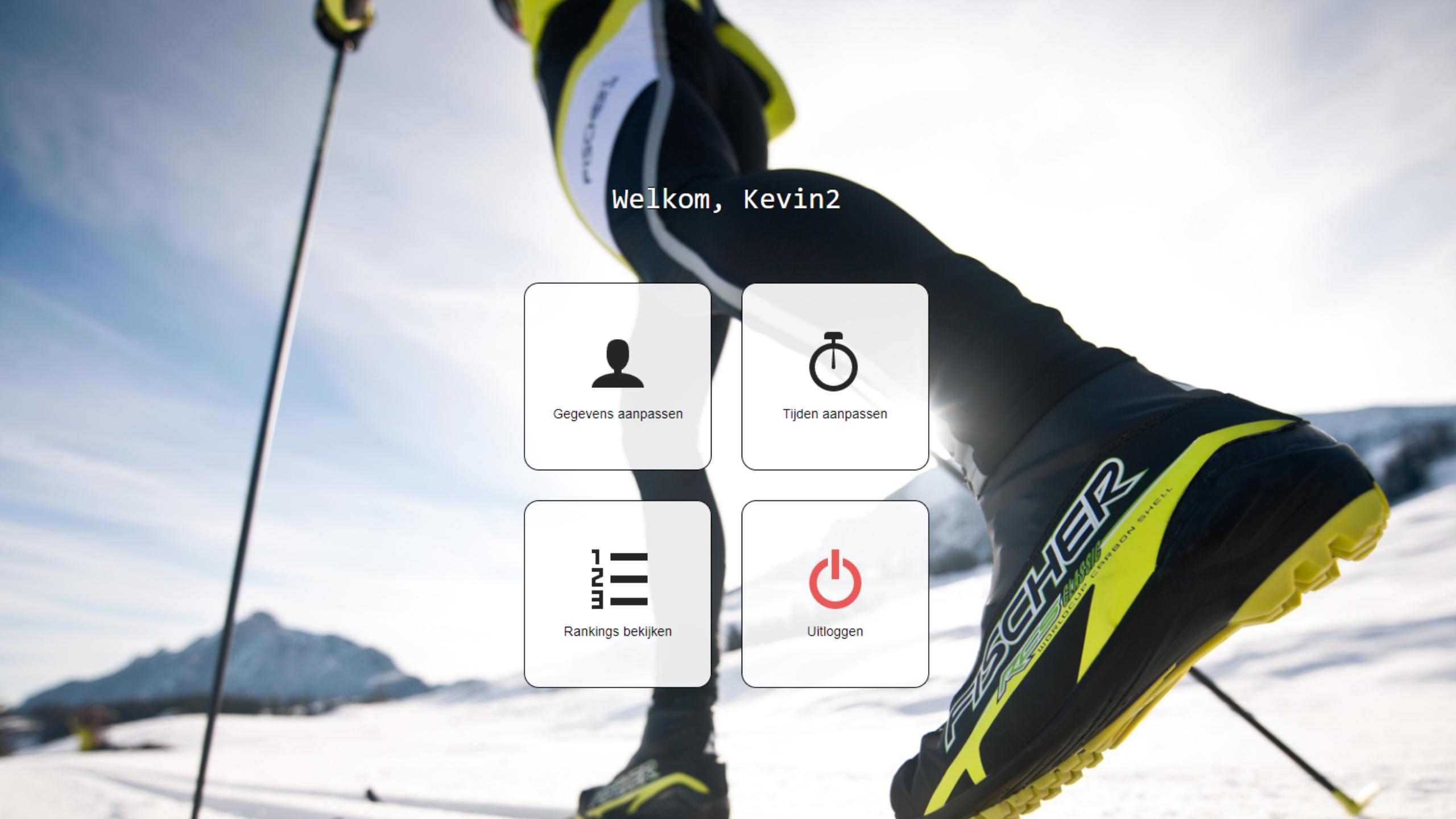
## Waarom heb ik informatica gekozen?

Ik heb informatica gekozen, omdat ik moest kiezen tussen biologie en informatica. De rede dat ik geen biologie heb gekozen is dat het mij totaal niet aansprak en ik er vanaf de eerste klas al niks mee had. Informatica daar tegen sprak me meer aan. Het leek me interessant hoe je sites maakte en hoe computers werken.

Uiteindelijk is dit een goede keuze geweest omdat ik zit te twijfelen tussen twee studies waarbij ik het bij beide nodig heb. Namelijk mechatronica en industrieel product ontwerper. Ook vind ik het gewoon erg leuk om een site zoals deze te maken.

Ik hoop dat u deze site HEEEEELLLLLL mooi vond en mij een goed punt gaat geven :).  
Ik heb er veel tijd ingestoken en heb het erg leuk gevonden om hem te maken!





Welkom, Kevin2



Gegevens aanpassen



Tijden aanpassen

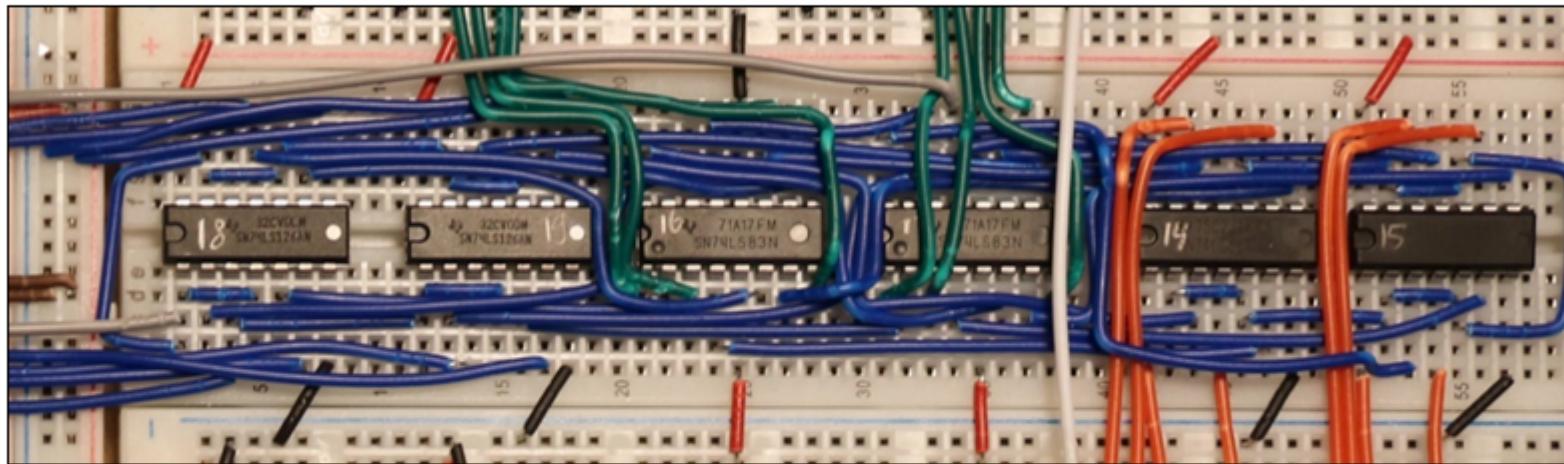


Rankings bekijken



Uitloggen

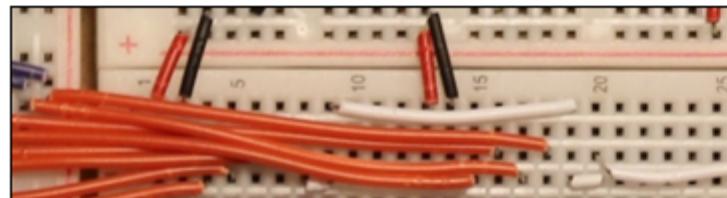
De adder ic's zijn niet in staat de uitkomst vast te houden. Zodra de input veranderd, zal de output meteen meeveranderen, net zoals dat in onze 4-bit adder ook gebeurt. Om te voorkomen dat hij de outputs meteen op de bus plaatst en alle andere opdrachten van de CPU interfereert, zijn er twee ic's die dit tegen houden: 18 en 19. Dit zijn weer 74LS126's, hetzelfde buffer-ic die we ook al gebruikt hebben in de A register. Deze houden dus de uitkomst vast, en kan deze laten zien zodra de controller erom vraagt.



Figuur 7

## B register

De B register is vrijwel hetzelfde als de A register, hoewel de B register niet in staat is de input terug te keren naar de BUS. Deze register gebruikt dus wel dezelfde register-ic's als de A register: de 74LS173. De data die erin vast zit kan eruitgehaald worden door de  $L_B$  aan te zetten. Deze data zal dan naar de ALU gaan, via de oranje kabeltjes (Figuur 8). Ook hier is de paarse draad voor de clock en de witte draad voor de  $L_B$ .



Ook kan het systeem rekening gaan houden met slaap ritmes van een gebruiker. Dit zijn natuurlijk dezelfde soort gegevens als het patroon van de gebruikers naar werk gaan patroon. Als iemand s 'nachts slaapt dan moeten er niet dezelfde dingen gebeuren als iemand gewoon wakker is, bijvoorbeeld de gebruiker wilt 's nachts een glas water pakken. Dan hoeft niet op de hele verdieping de lampen aan te springen. Maar eigenlijk een klein deel van de lampen om de gebruiker door de gangen te helpen, je kunt dit ook combineren met led lampen, zo kun je een heel zacht dimlicht aanzetten om iemand naar de bad kamer te begeleiden. Op dit soort zaken kun je inspelen omdat je weet dat de rest van het huis aan het slapen is en dat niet iedereen wakker hoeft te worden.

het is dus voor een slimme lamp ook belangrijk om de patronen van zijn gebruiker te volgen zodat de computer slim kan inspelen op wat de gebruiker fijn vindt.

Nadat je al deze gegevens hebt om te kunnen bepalen wanneer de lamp aan en uit moet. Is het natuurlijk ook belangrijk dat ie over de juiste hardware beschikt om de lamp mee aan te



---

### *De Hardware opstelling*

---