

Week 2 Opgave 3

(c) 2019 HvA.nl, f.h.schippers@hva.nl; versie 1.0

Bit-adders

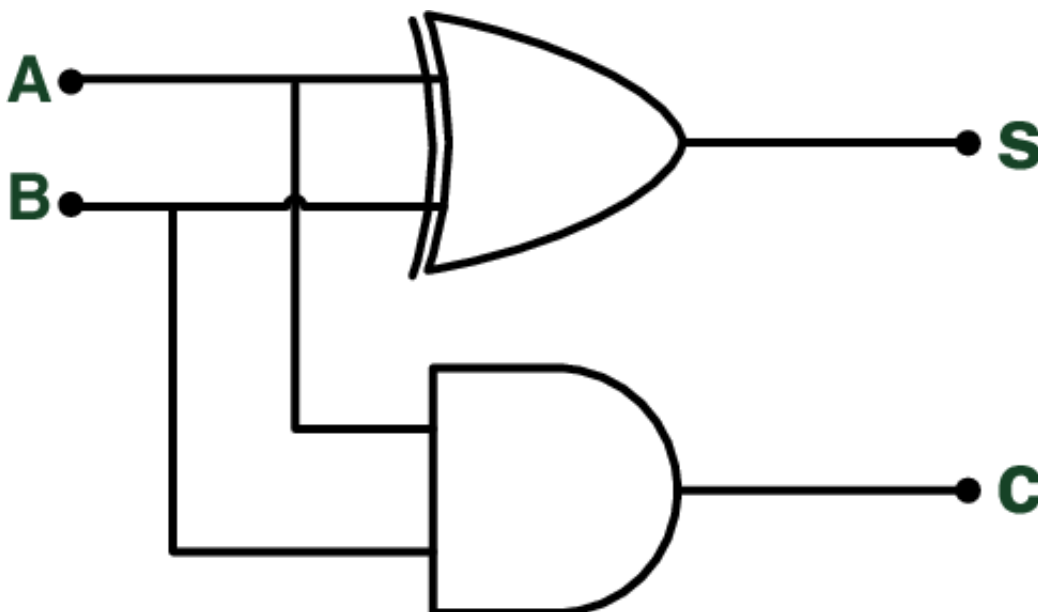
In deze opgaven van week 2 simuleren we gates, circuits, half-adder, full-adder en uiteindelijk een adder voor een register met n-bits.

De opgave `w2o4` is een uitbreiding op deze opgave. Zorg dus dat de tests slagen voordat je ze gebruikt in de volgende opgave.

Adders tellen de input bits op geven een som en een carry.

Half-adder

Een half-adder telt twee bits (`iBitA` , `iBitB`) op tot een som (`oBitS`) en een carry (`oBitC`) De file `w2o3.py` bevat the skeleton code om een half-adder te maken.



Half-adder

```
1 class HalfAdder(Circuit):
2     def __init__(self, iBitA, iBitB, oBitS, oBitC):
3         Circuit.__init__(self)
4         # ... write your code
```

- 1 De definitie van `HalfAdder` een sub-klasse van `Circuit`
- 2 De methode methode `__init__` bepaald in input- en output-bits.

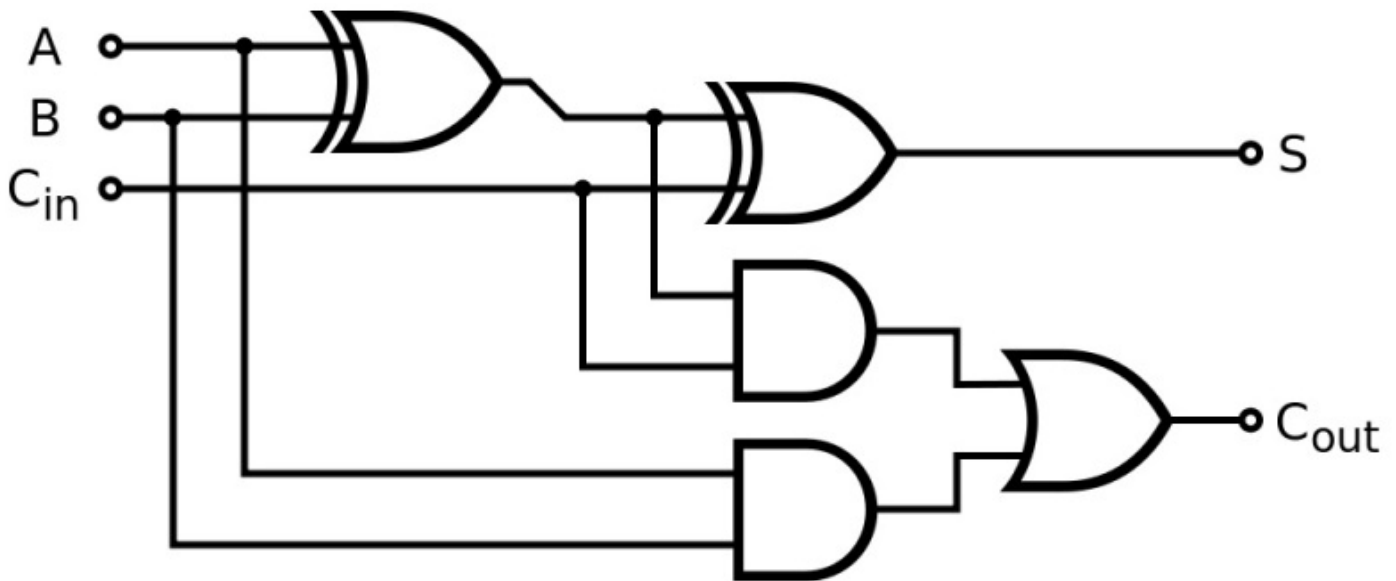
3 Hier roepen we de `__init__` methode van de super-klasse aan. Dit zorgt er voor dat de super-klasse goed geïnitiliseerd wordt.

4 Voeg de juiste code toe.

Full-adder

Een full-adder telt drie bits (`iBitA` , `iBitB` , `iBitC`) op tot een som (`oBitS`) en een carry (`oBitC`) De file `w2o3.py` bevat the skeleton code om een full-adder te maken.

Full-adder



```
1 class HalfAdder(Circuit):
2     def __init__(self, iBitA, iBitB, iBitC, oBitS, oBitC):
3         Circuit.__init__(self)
4         # ... write your code
```

Opgave 3

Adders

De opdracht is om de half-adder en de full-adder te implemeteren. Je mag andere gates dan de Nand-gate ook gebruiken

Inleveren

Zorg dat je naam in het programma staat (`__author__`) en lever jouw versie van `w2o3.py` in op

VLO. Inleveren voor **vrijdag 15 feb 24:00** (voltijd) of **maandag 18 feb 24:00** (deeltijd).