

Week 2 Opgave 4

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

Gates, Circuits, Adders

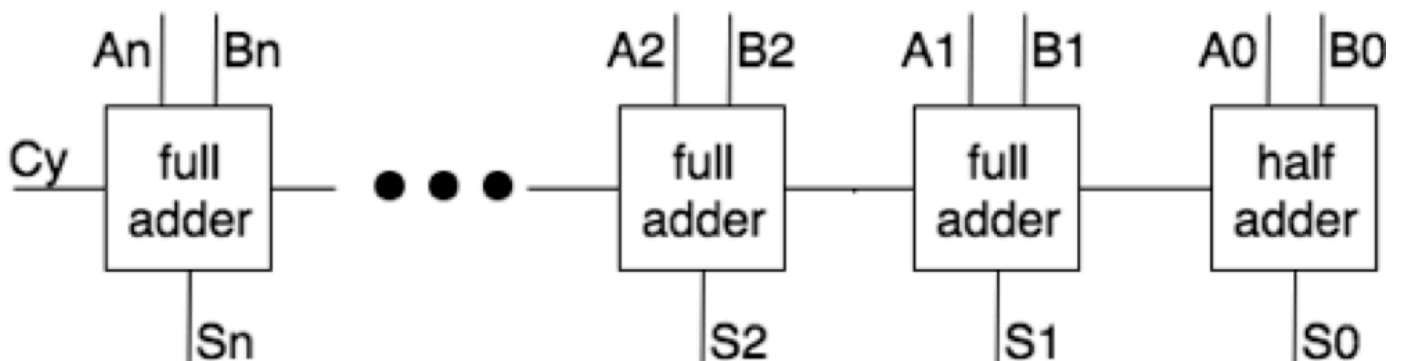
In deze opgaven van week 2 simuleren we gates, circuits, half-adder, full-adder en uiteindelijk een adder voor een register met n-bits.

N-Bit-Adder

Een n-bit-adder telt twee n-registers op in een derde n-register. We maken gebruik van een half-adder en full-adders om de optelling te realiseren.

Het schema van een n-bit adder is als volgt:

n-bit adder



De definitie van de class begint met:

```
1 class Adder(Circuit):
2     def __init__(self, iRegA, iRegB, oRegC, n=4):
3         Circuit.__init__(self)
4         # ... write your code
```

Net als met de half- en full-adder hoeft alleen de `__init__` geïmplementeerd te worden.

Bekijk in `w2_lib.py` hoe een register is geïmplementeerd. Wat doe onderstaande code?

```
>>> from w2_lib import *
>>> regA = Register.fromStr(bitStr="00001101", name="A")
>>> int(regA[-1])
# Wat is hier het resultaat
>>> int(regA[-8])
# Wat is hier het resultaat
```

Denk aan de opgave `w1o2` . Dat was ook een n-bit-adder.

Opgave 4

De file `w2o3.py` bevat de skeleton-implementatie van een n-bit-adder. De opdracht is om de n-bit-adder verder te implementeren.

Er zijn twee test in `w2o4.py` . Waarom geven deze andere maar juiste antwoorden?

Inleveren

Zorg dat je naam in het programma staat (`__author__`) en lever jouw versie van `w2o4.py` in op VLO. Inleveren voor **vrijdag 15 feb 24:00** (voltijd) of **maandag 18 feb 24:00** (deeltijd).