

Hoang Long, Vu - 231688  
Duc Huy, Nguyen - 231611

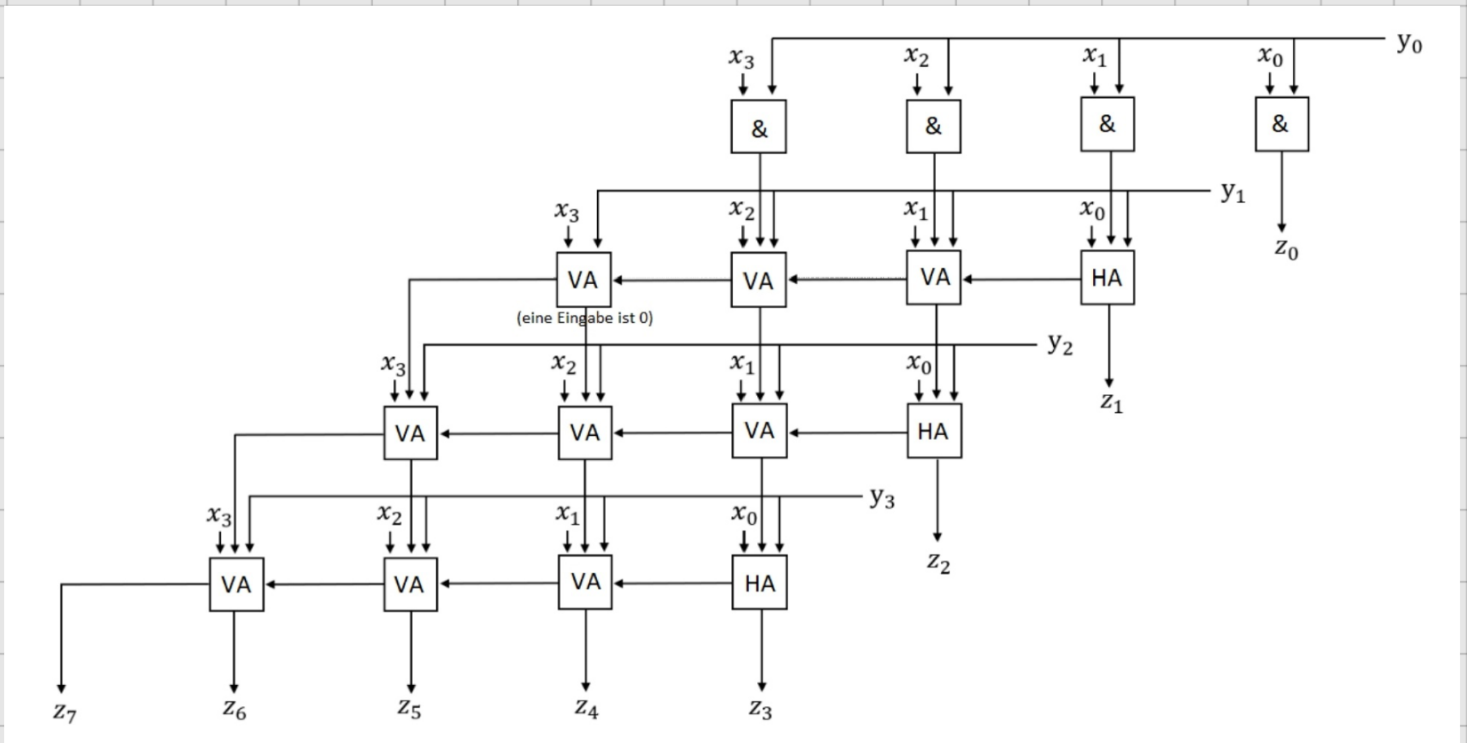
## Aufgabe 1

a)

Multiplizieren  $\rightarrow$  gegeben Block1 (UND-Gatter)

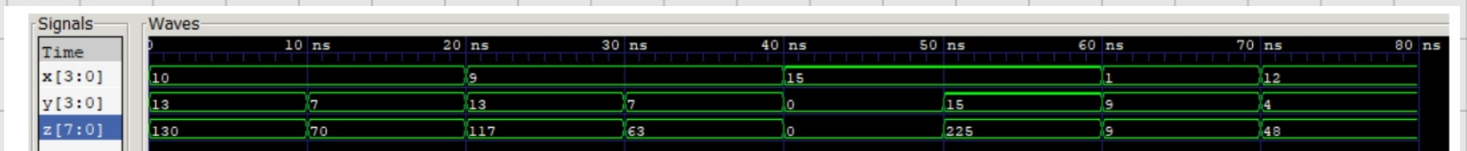
Addieren mit vor. Übertrag  $\rightarrow$  gegeben Block2 (VA)

Addieren ohne vor. Übertrag  $\rightarrow$  gegeben Block3 (HA)



b) im code.

c) Mit erweiterte Testergebnisse:



## Aufgabe 2:

a) im Code

b).

### c) Add-shift-Multiplexer

- verwendet eine schrittweise Methode.
- multipliziert den Multiplikator mit jedem Bit des Multiplikanden und addiert die Teilergebnisse schrittweise, indem er die Bits des Multiplikators nach links schiebt.

### Parallel-Multiplexer

- verwendet eine parallele Methode
- multipliziert den Multiplikator mit allen Bits des Multiplikanden gleichzeitig und addiert die Teilergebnisse in einem Schritt.

### Add-shift-Multiplexer

#### Vorteil

- einfache Implementierung: nur Additionen und Shift-Operationen verwendet
- geringerer Hardwarebedarf: im Vergleich zum Parallel-Multiplexer

#### Nachteil

- langsam, große Berechnungszeit: da er schrittweise erfolgt, benötigt mehr Takte
- Höhere Latenz: das Ergebnis erst später verfügbar ist.

### Parallel-Multiplexer

#### Vorteil

- schneller: da er parallel erfolgt
- niedrigere Latenz: das Ergebnis schneller verfügbar ist.

#### Nachteil

- Komplexere Implementierung: er erfordert eine komplexere Schaltung
- Höherer Hardwareaufwand.
- Höherer Stromverbrauch: Aufgrund der Verwendung einer größeren Anzahl von Schaltelementen

d) eine Möglichkeit zur Optimierung : verwendet Carry-lookahead-Addierer, um schnell die Summe von mehreren Teilprodukten zu berechnen.