57107
53704

Vorbereitung Versuch 1

① a) Man kann die Zeiten direkt eingeben oder mit Hilfe Funktion (Schleife).

b) In der Chip-Report Datei findet man das „Chip Diagram".

②

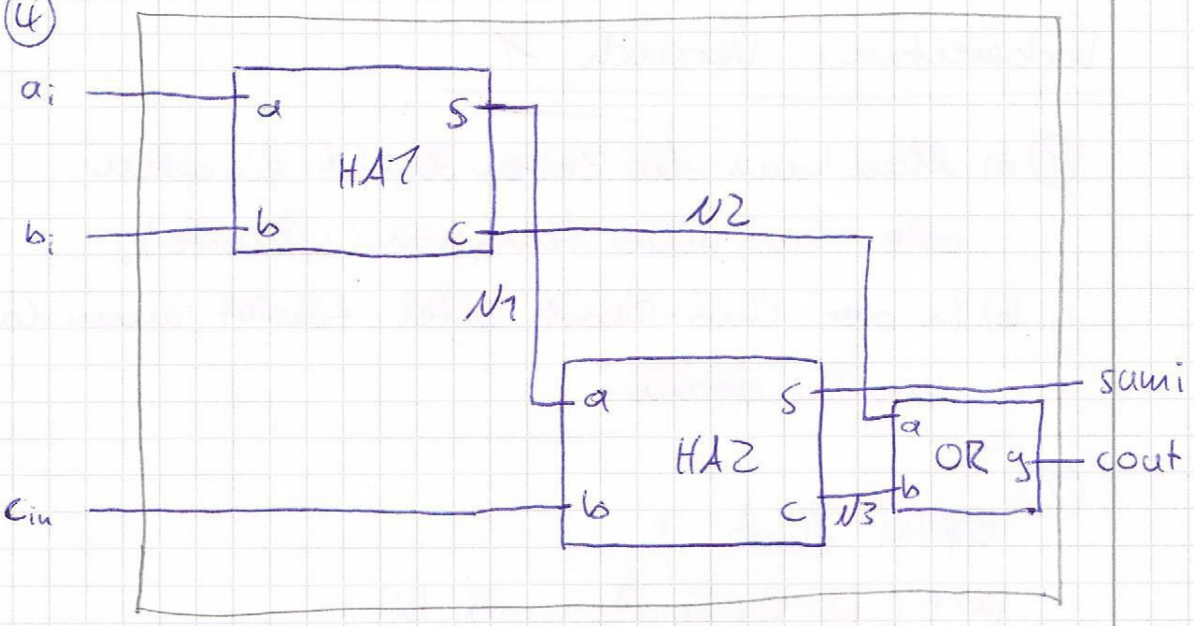
```
entity fktab is
port ( D, C, B, A : in bit;
                Y : out bit);
end;

architecture demo of fktab is
begin
with bit_vector'( D, C, B, A ) select
    Y <= '1' when "0000",
         '1' when "0101",
         '1' when "0110",
         '1' when "1001",
         '1' when "1010",
         '1' when "1111",
         '0' when others;
end demo;
```

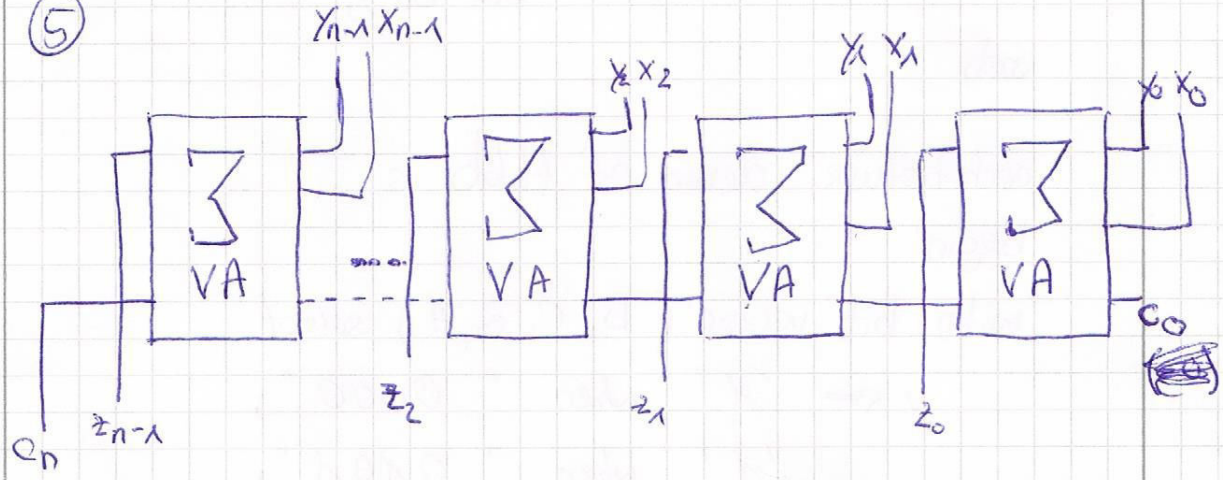
③

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

④

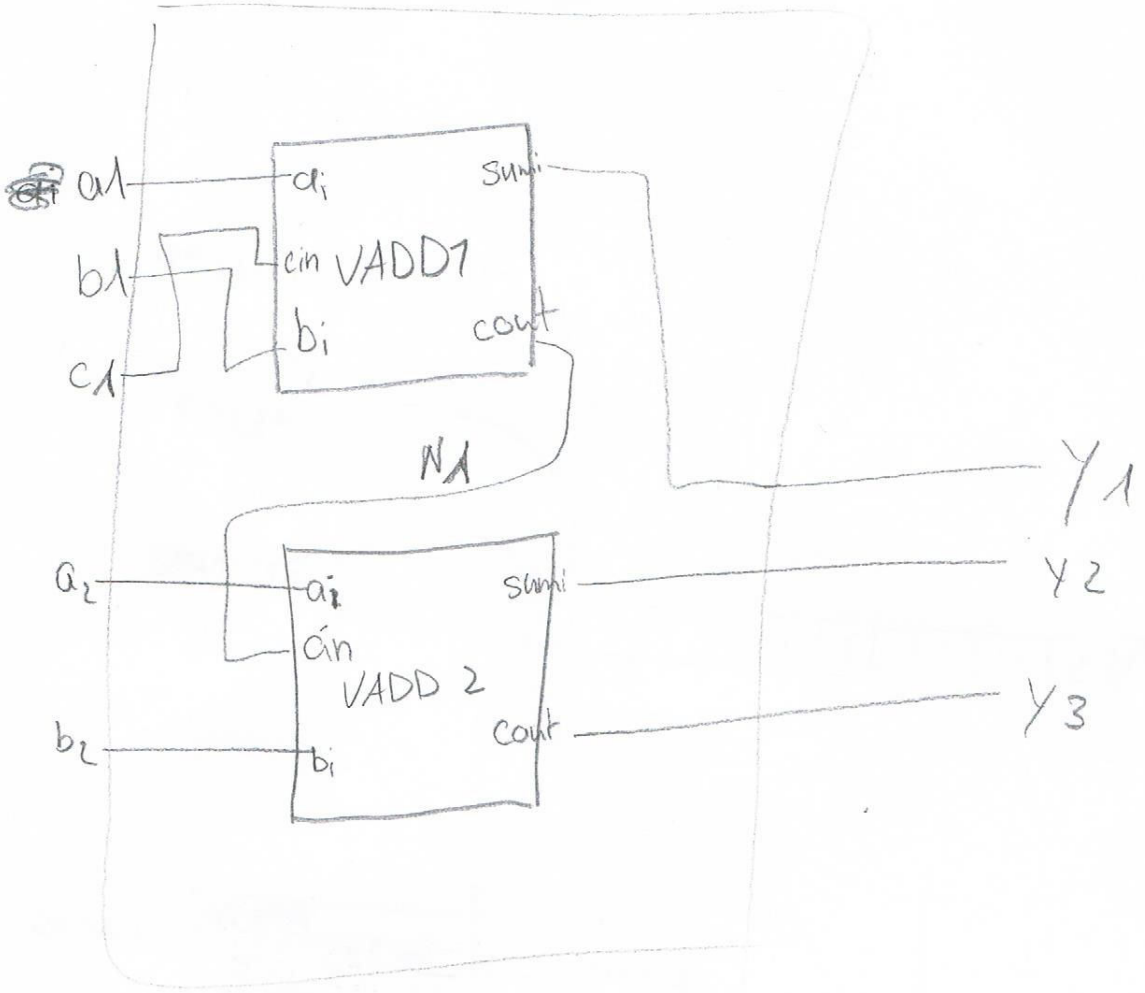


⑤



- Summe hat ein Bit mehr als die Summanden aufgrund des Übertragbits.
- Durch die obersten 2 Bits der Eingangszahlen (x_{n-1}, x_{n-1}).

Skizze Serienaddierer mit 2 Volladdierern:



Digitallabor – Versuch 1

Versuch: Kombinatorisches und strukturelles VHDL im GAL Baustein

Aufgabe 1

Aufgabe war es ein nebenläufiges VHDL Modell für ein ODER Gatter mit zwei Eingängen zu schreiben zu programmieren und dann zu Testen. Durch einfache logische oder-Verknüpfung unserer Eingänge A und B in dem VHDL Programm konnten wir unser ODER Gatter beschreiben. Nachfolgend wird gezeigt wie das ODER Gatter realisiert worden ist.

```
architecture ODER2_arch of ODER2_ent is
begin
        y <=  a OR b;
end ODER2_arch;
```

Nach Programmierung des Bausteins wurde dieser mit zwei Schaltern und einer LED getestet. Mit Hilfe von einfachem Durchschalten der oder-Verknüpfung konnte dann festgestellt werden, ob der Baustein wie gewünscht funktioniert.

Aufgabe 2

Ziel war es ein nebenläufiges VHDL Modell für einen Halbaddierer mithilfe einer Funktionstabelle zu schreiben. Die Funktionstabelle wurde in VHDL realisiert. Es wurden zwei Funktionstabellen geschrieben. Eine für die Summe s und eine für den Übertrag c. Nach testen des Designs mit Hilfe der Simulation stellte sich heraus, dass die Funktionstabellen und somit der Halbaddierer richtig beschrieben worden ist.

Aufgabe 3

Mithilfe der vorgefertigten Instanzen des Halbaddierers und des ODER Gatters aus den vorherigen Aufgaben, sollte nun eine Zusammenstellung dieser erfolgen um einen Volladdierer zu realisieren. Wir erstellten ein neues VHDL Projekt in dem wir die Komponenten des Halbaddierer und die Komponenten des ODER Gatters verknüpften. Die Komponenten wurden realisiert, indem die entity jeweils kopiert wurde und dann ‚entity' und ‚end' im Quelltext durch ‚component' und ‚end component' ersetzt wurden.
Wir haben die Skizze aus der Versuchs-Vorbereitung (Aufgabe 4) benutzt und sie um die Beschriftung der Leitungen N1 bis N3 ergänzt um die Port Map des Volladdierers zu ertellen. Durch simulieren des Volladdierers wurde die Richtigkeit der Funktionsweise überprüft.

Aufgabe 4

Aufgabe war es einen Serienaddierer für 2 Zahlen zu realisieren. Wir benutzten dasselbe Verfahren wie in Aufgabe 3, nur das wir anstatt Halbaddierer und einem ODER Gatter, zwei Komponenten des Volladdierers benutzt haben. Zur Hilfe benutzten wir eine Skizze. Anschließend testeten wir es mit Hilfe einer Testbench und programmierten es nach erfolgreicher Simulation auf unseren Baustein. Anschließend testeten wir den Baustein an unserer Vorrichtung mit einer 7-Segment-Anzeige und kamen zu dem Ergebnis das zwei plus drei gleich fünf ergibt.