1. Übung	**M9 VLSI Anwendungen** Hochschule für Technik und Wirtschaft	SS 2021

Ausgabe: 15./22. April 2021 Abgabe: 29. April / 5. Mai 2021

Schwerpunkte

- Arithmetische Grundfunktionen
- Halb- und Volladdierer
- Carry Select Addierer

Aufgabe 1

Im Rahmen der 1. Übung soll für eine generische Addierer-Entity `m9_adder` eine Architektur `carry_select` entworfen und implementiert werden.

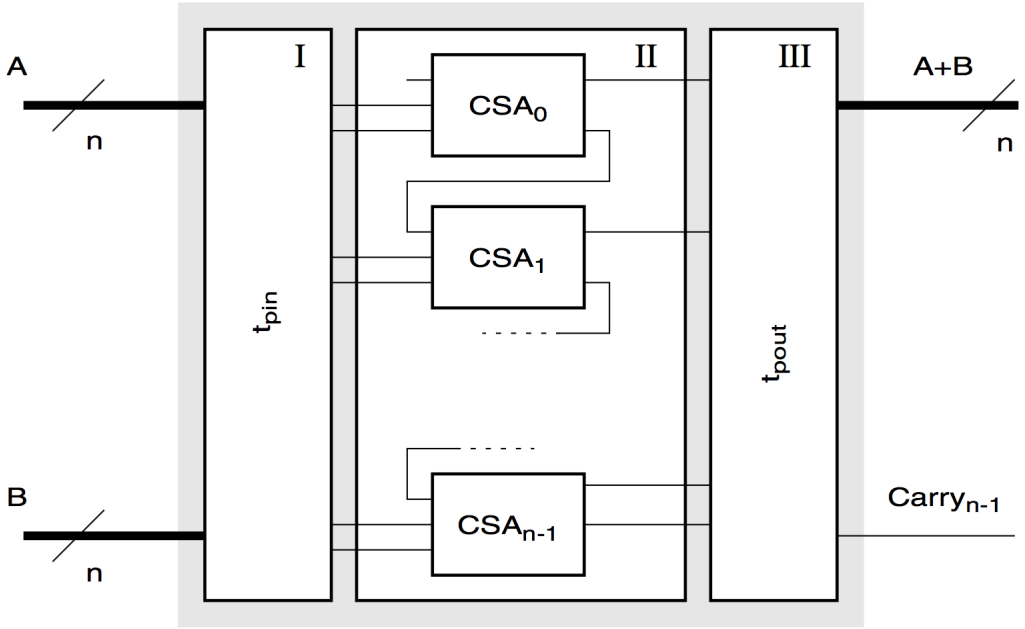

Abbildung 1: Struktur der `carry_select` Architektur des generischen Addierers.

Wie Abbildung 1 zu entnehmen ist, unterteilt sich die Architektur in drei funktionelle Blöcke: Block II realisiert einen Carry-Select-Addierer; eingangsseitige Verzögerungen können durch Block I nachgebildet werden. Sollte die verwendete Architektur der Carry-Select Addierer aus Block II keine Timing-Aspekte berücksichtigen, könnte das designtypische “Durchpropagieren” der Überträge ebenfalls in Block III als resultierende Gesamttransportverzögerung nachgebildet werden.

Die Struktur einer einzelnen Carry-Select Einheit können Sie Abbildung 2 entnehmen. Dabei soll auf die bereits implementierte Komponente `m9_fulladd` (welche `m9_halfadd` verwendet) aus den elektronischen Unterlagen zurückgegriffen werden.

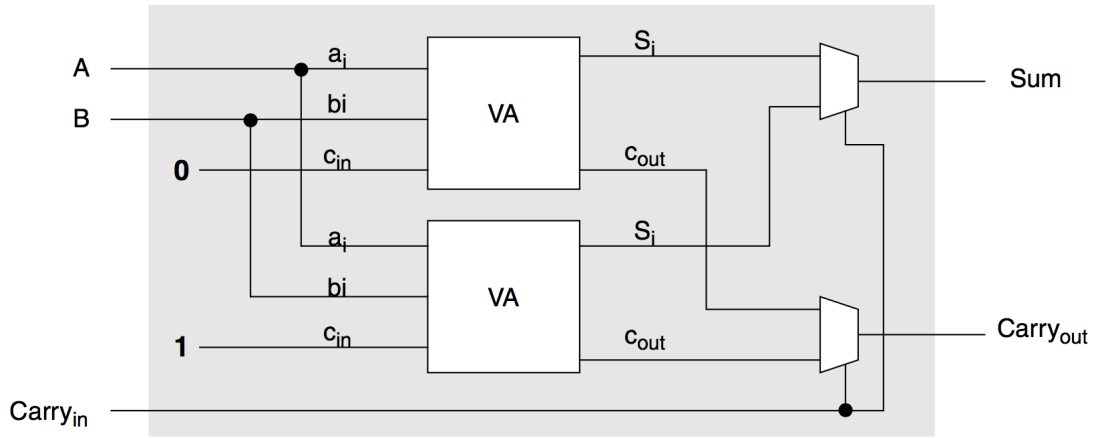

Abbildung 2: Gatelevel-Architektur einer Carry-Select Einheit

Aufgabe 1.1

Im folgenden soll nun das eingangs in Abbildung 1 vorgestellte Design umgesetzt werden. Der Addierer weist folgende Entity-Spezifikation auf:

```
entity m9_adder is
  generic (width : positive := 5);
  port (cin    : in  std_logic;
        a, b   : in  std_logic_vector (width - 1 downto 0);
        s      : out std_logic_vector (width - 1 downto 0);
        cout   : out std_logic);
end m9_adder;
```

Realisieren die oben genannte Architektur `carry_select` eines bzgl. seiner Summandenbreiten parametrisierten Addierers unter Verwendung der generate-Anweisung. Beschränken Sie sich in ihrer Implementierung auf die reine Funktionalität des Blockes II. Testen Sie ihr Design mit ein paar geeigneten Testvektoren.

Aufgabe 1.2

Die obigen Halb - und Volladdierer aus den elektronischen Unterlagen modellieren keinerlei Verzögerungen. Wie könnte eine Architektur `carry_select_delayed` aussehen, die eingang- und ausgangsseitige Transportverzögerungen (Block I und III) nachbildet?

Aufgabe 1.3

Synthetisieren Sie Ihre CSA-Schaltung mit Vivado für die Xilinx Zynq Z7010 FPGAs des ZedBoards und ermitteln Sie die Laufzeiteigenschaften der Schaltung bei einer Operandenbreite von 32 Bit.