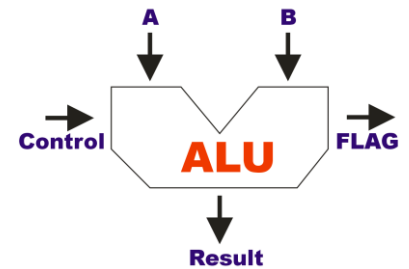




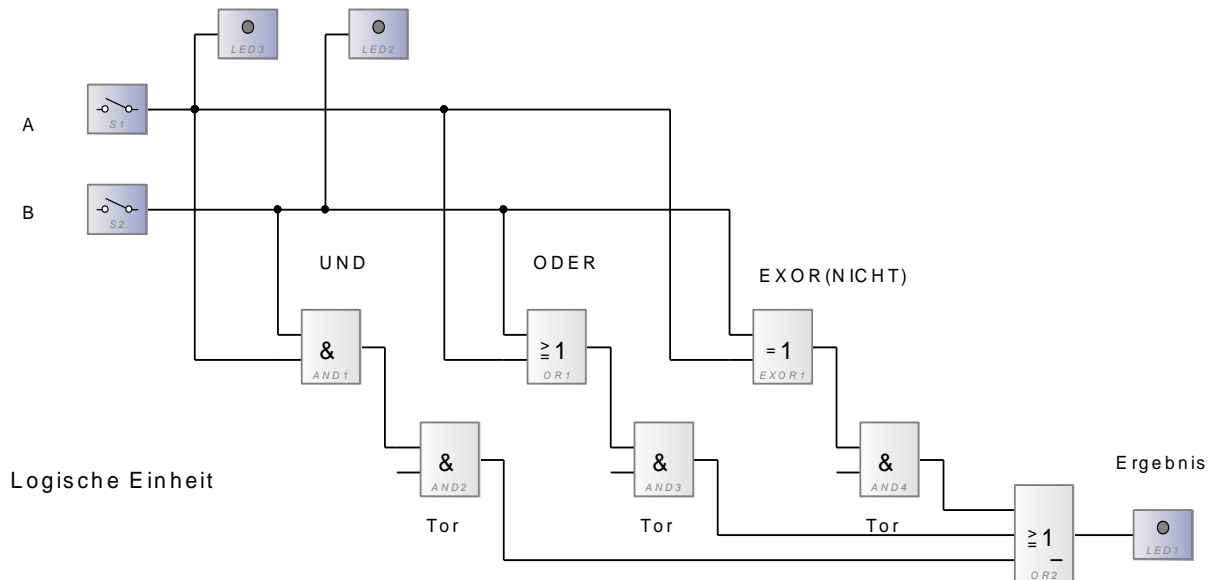
Eine arithmetisch-logische Einheit (englisch: arithmetic logic unit, daher oft abgekürzt ALU) ist ein elektronisches Rechenwerk, welches in Prozessoren zum Einsatz kommt. Die ALU berechnet arithmetische und logische Funktionen. Sinnvollerweise kann sie mindestens folgende Minimaloperationen durchführen:

- **Arithmetisch:** Addition (ADD)
- **Logisch:** Konjunktion (Und-Verknüpfung, AND)
Disjunktion (Oder-Verknüpfung, OR)
Negation (Nicht-Verknüpfung, NOT)

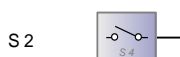



Eine ALU kann meistens zwei Binärwerte (hier A und B) mit gleicher Stellenzahl (n) miteinander verknüpfen. Man spricht von n-Bit ALUs. Typische Werte für n sind 8, 16, 32 und 64. Die n-Bit ALU ist meist aus einzelnen 1-Bit ALUs zusammengesetzt. Die Auswahl der durchzuführenden Verknüpfung erfolgt durch die Steuereingänge (Hier S1 und S2).

1-Bit ALU



Auswahl der Funktion



 Friedrich-August Haselwander Gewerblich-Technische Schulen Offenburg	Programmierbare Logik	Klasse:
	Arithmetic Logic Unit	Datum:

Aufgaben:

1. Überlegen Sie sich, wie die Anwahl der Funktionen realisiert werden kann und zeichnen Sie die sich ergebende Logik in die Schaltung mit ein.
Hinweis: orientieren Sie sich am Aufbau der unten stehenden Wahrheitstabelle.
2. Tragen Sie festgelegten **logischen** Verknüpfungen in die Felder bei der Wahrheitstabelle ein.
3. Vervollständigen Sie die Wahrheitstabelle für das Ergebnis E für die **logischen** Verknüpfungen.
4. Setzen Sie die Wahrheitstabelle mit Hilfe von „ispLEVER Project Navigator“ in ein ABEL-Programm (TRUTH_TABLE) für die PLD-Hardware um.
Hinweis: Legen Sie vor dem Erstellen des Projektes einen neuen Ordner dafür an. Achten Sie darauf, dass sie als Ziel den LC4128V aus der ispMach4000 und als Typ ABEL verwenden.
5. Testen Sie ihr Programm mit Hilfe der TEST_VECOTRS und auf der Hardware.

Aus Aufgabe 1 ist noch eine Kombination frei. Diese soll nun dazu verwendet werden, die Arithmetische Einheit zu realisieren. Diese soll die Funktion des Addierens erfüllen. Das Ergebnis soll mit Hilfe von Ergebnis E und dem Übertrag (Carry Out) Cout ausgegeben werden.

6. Vervollständigen Sie die Wahrheitstabelle für das Ergebnis E und Übertrag Cout für die arithmetische Verknüpfung.
7. Speichern Sie ihr aktuelles Projekt unter einem neuen Namen in einen zweiten Ordner. Ergänzen Sie nun das neue Projekt im Programm mit der Arithmetischen Einheit.
8. Testen Sie ihr erweitertes Programm auf der Hardware.
9. Zeichnen Sie die sich ergebenden Verknüpfungen für die Arithmetisch Einheit in die obige Schaltung mit ein.



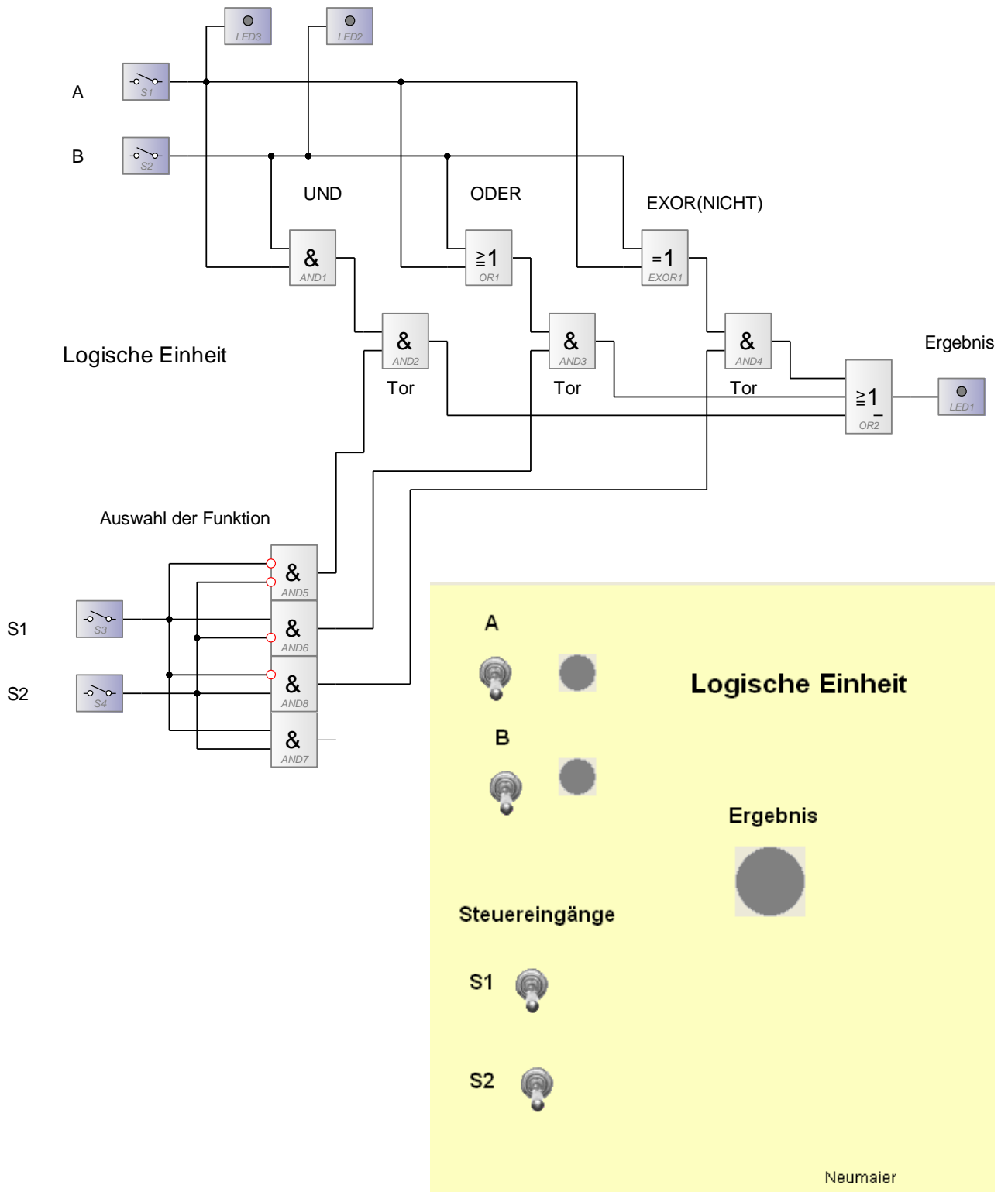
Wahrheitstabelle für die 1 Bit Arithmetisch-Logische Einheit

	S2	S1	B	A	E	Cout	
[]	0	0	0	0			} Logik-Verknüpfungen
	0	0	0	1			
	0	0	1	0			
	0	0	1	1			
[]	0	1	0	0			
	0	1	0	1			
	0	1	1	0			
	0	1	1	1			
[]	1	0	0	0			
	1	0	0	1			
	1	0	1	0			
	1	0	1	1			
[]	1	1	0	0			
	1	1	0	1			
	1	1	1	0			
	1	1	1	1			



Lösungen:

Logische Einheit





MODULE LOGIC

TITLE '1Bit Logische Einheit'

"-----

" Programm zur Umsetzung 1Bit Logische Einheit

" "-----

DECLARATIONS

"-----

" Eingänge: S1, S2, A und B: Steuereingänge und Variablen

"-----

S1, S2, A, B PIN 19, 20, 21, 22;

"-----

" Ausgänge: Ergebnis E

"-----

E PIN 53 ISTYPE 'COM';

"istype 'com' gibt an, dass die Ausgänge statisch sind.

"Sie haben kein Speicherverhalten

"-----

" Zuordnung der Eingangsgrößen zum Ausgang

"-----

TRUTH_TABLE

([S2, S1, B, A] -> [E]); //Anlegen einer Wahrheitstabelle

[0, 0, 0, 0] -> [0];

[0, 0, 0, 1] -> [0];

[0, 0, 1, 0] -> [0];

[0, 0, 1, 1] -> [1];

[0, 1, 0, 0] -> [0];

[0, 1, 0, 1] -> [1];

[0, 1, 1, 0] -> [1];

[0, 1, 1, 1] -> [1];

[1, 0, 0, 0] -> [0];

[1, 0, 0, 1] -> [1];

[1, 0, 1, 0] -> [1];

[1, 0, 1, 1] -> [0];

[1, 1, 0, 0] -> [0]; // Steuereingänge 1 1 Nicht belegt

[1, 1, 0, 1] -> [0];

[1, 1, 1, 0] -> [0];

[1, 1, 1, 1] -> [0];

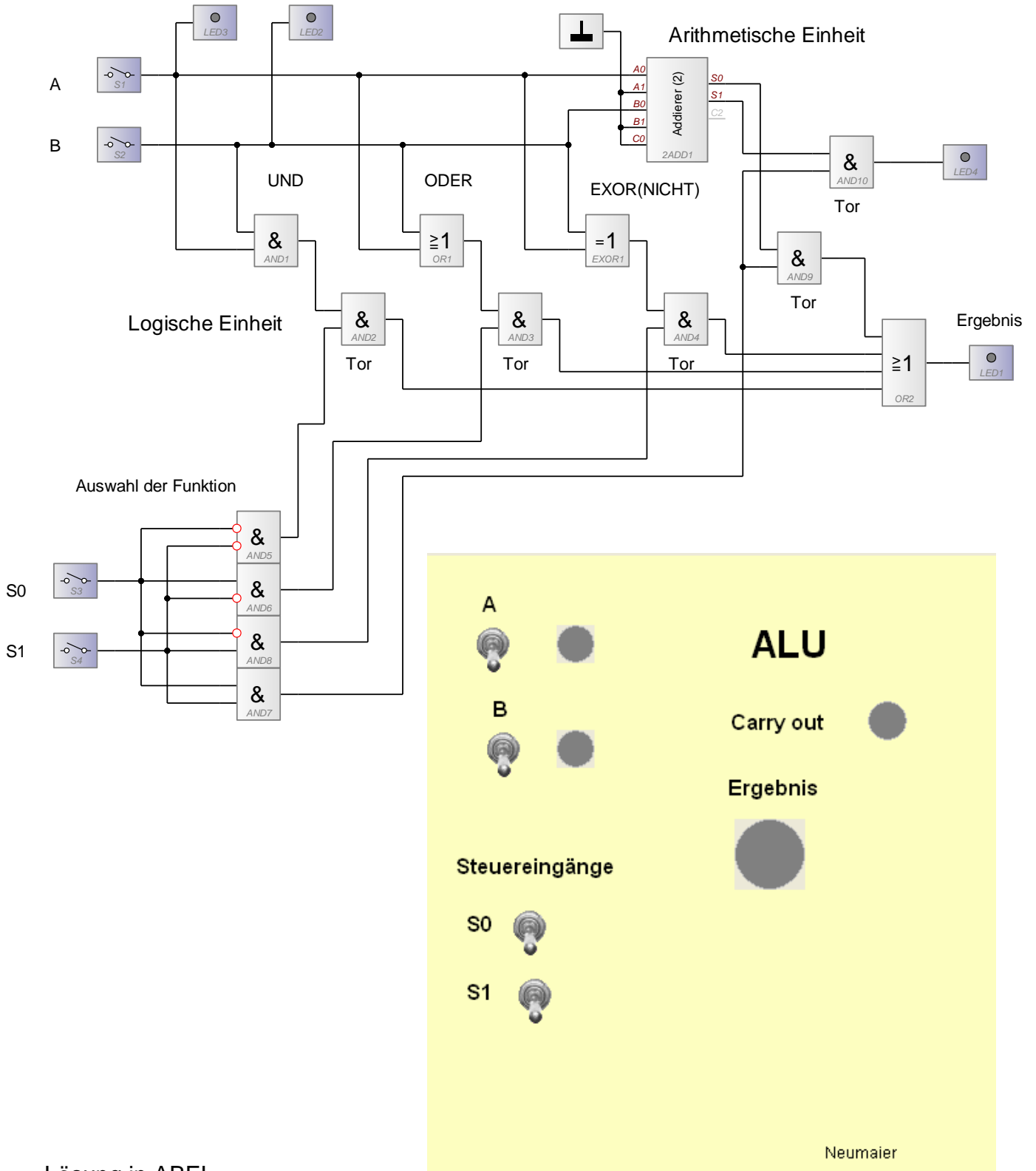
"Hinweis: Strichpunkt am Ende der Befehlszeile nicht vergessen!!

END



Arithmetik Logic Unit

ALU (Arithmetik Logic Unit)



Lösung in ABEL



MODULE ALU

TITLE '1bit_ALU'

Declarations

"Eingaenge definieren

"Programiereingaenge haben den Namen S1 und S0

S1, S0 pin 3,2; //Port1 D0 und D1

E1, E0 pin 6,5; "Variablen, jeweils 1 bit

"Ausgaenge definieren

Cout, Ausg pin 14,15, istype 'com';

"istype 'com' gibt an, dass die Ausgaenge statisch sind

truth_table //Wahrheitstabelle
([S1, S0, E1, E0]->[Ausg, Cout]);

"UND" [0, 0, 0, 0]->[0, 0];
[0, 0, 0, 1]->[0, 0];
[0, 0, 1, 0]->[0, 0];
[0, 0, 1, 1]->[1, 0];

"ODER" [0, 1, 0, 0]->[0, 0];
[0, 1, 0, 1]->[1, 0];
[0, 1, 1, 0]->[1, 0];
[0, 1, 1, 1]->[1, 0];

"exOR" [1, 0, 0, 0]->[0, 0];
[1, 0, 0, 1]->[1, 0];
[1, 0, 1, 0]->[1, 0];
[1, 0, 1, 1]->[0, 0];

"+" [1, 1, 0, 0]->[0, 0];
[1, 1, 0, 1]->[1, 0];
[1, 1, 1, 0]->[1, 0];
[1, 1, 1, 1]->[0, 1];

END