

# ALU

*Beim sogenannten Ripple-Carry-Adder durchläuft das Carry-Signal alle Einzel-Addierer. Dies bedingt eine große Verzögerungszeit, bis das endgültige Ergebnis feststeht. Suchen Sie in der Literatur nach schnelleren Addierern.*

*Antwort:*

# ALU

*Entwerfen Sie ein (einfaches) Schaltnetz, das eine 8-Bit-Zahl inkrementieren kann.*

*Antwort:*

# ALU

*Entwickeln Sie ein (einfaches) Schaltnetz, das mit 10 multiplizieren kann! Als Input sind 4-Bit-breite Operanden vorgesehen. Volladdierer dürfen als bekannt vorausgesetzt und als Bauteil verwendet werden. (Ehemalige Klausuraufgabe.)*

**Antwort:**

# ALU

*Setzen Sie Gatter (AND, OR, NOT) zusammen, so dass sich ein Schaltnetz mit folgendem Ausgabe-Verhalten ergibt (Multiplexer):*

X	Y	M		Z
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		1
1	0	1		0
1	1	0		1
1	1	1		1

ist  $M=0$ , ergibt sich  $Z=X$

ist  $M=1$ , ergibt sich  $Z=Y$

**Antwort:**

# ALU

*Welche Einheit der ALU wird für den Assemblerbefehl `cmp` benötigt?*

*Wie können die Befehle `ja/jb` (unsigned, above/below) sowie `jge/jle` (signed, greater or equal/lesser or equal) feststellen, ob gesprungen werden soll oder nicht? Entwerfen Sie jeweils eine kleine Schaltung, die genau dann eine '1' liefert wenn gesprungen werden soll.*

# ALU

*Entwickeln Sie ein Schaltnetz für einen 2-Bit Multiplizierer!*