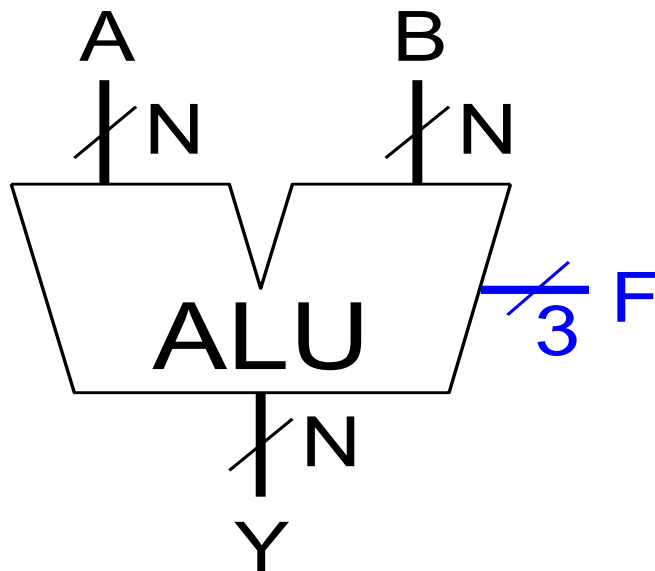
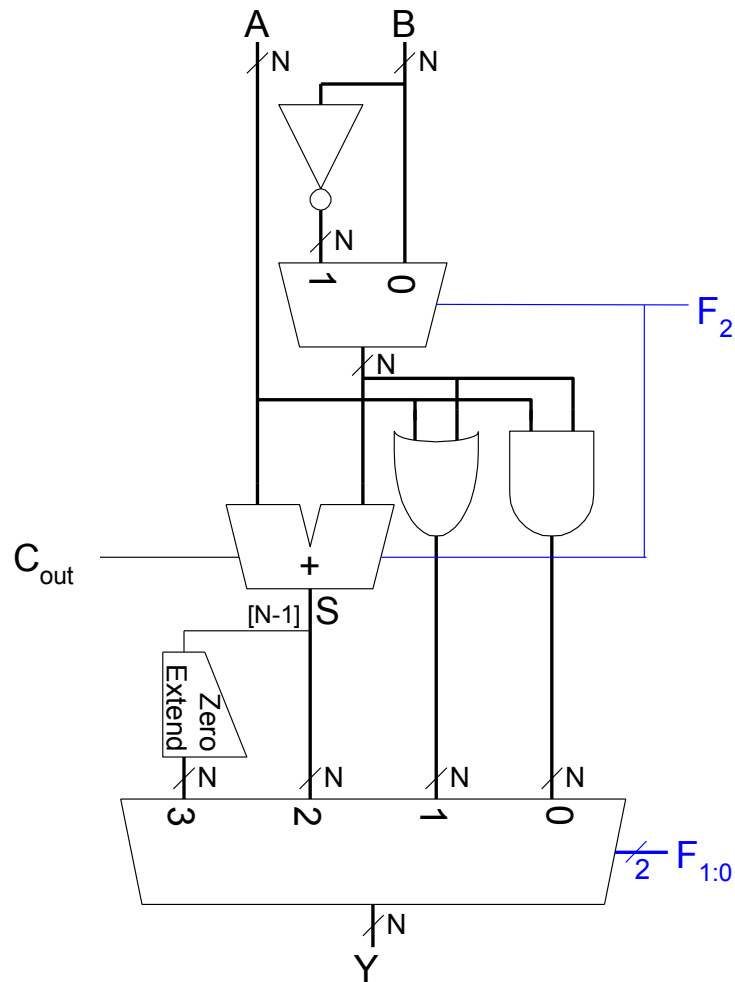


Arithmetisch-logische Einheit (*arithmetic logic unit, ALU*)



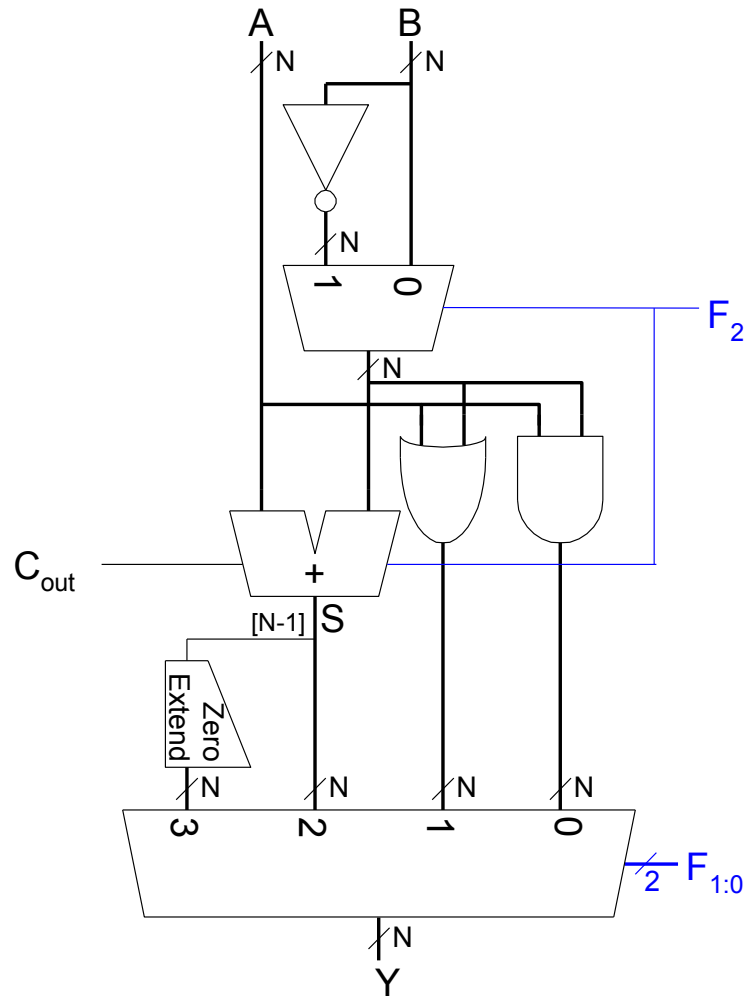
$F_{2:0}$	Funktion
000	$A \& B$
001	$A B$
010	$A + B$
011	Nicht verwendet
100	$A \& \sim B$
101	$A \sim B$
110	$A - B$
111	SLT

Entwurf einer ALU



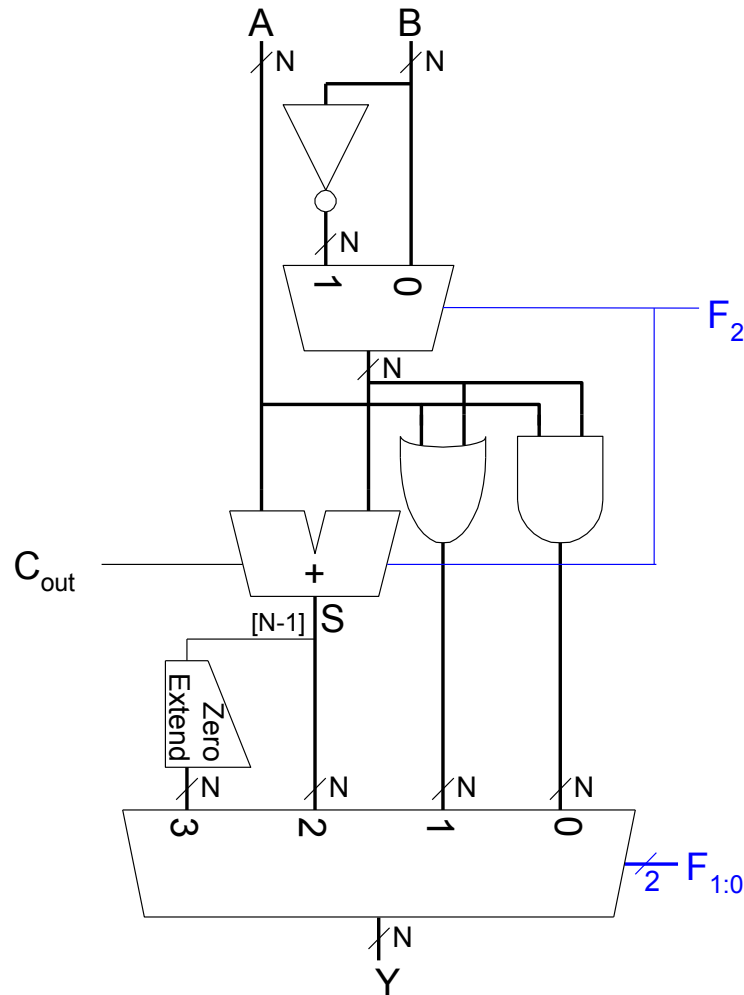
$F_{2:0}$	Funktion
000	$A \& B$
001	$A B$
010	$A + B$
011	Nicht verwendet
100	$A \& \sim B$
101	$A \sim B$
110	$A - B$
111	SLT

Beispiel: Set Less Than (SLT)



- Konfiguriere 32b ALU für SLT-Berechnung
- Annahme: $A = 25$, $B = 32$

Beispiel: Set Less Than (SLT)



- Konfiguriere 32b ALU für SLT-Berechnung
 - Annahme: $A = 25$, $B = 32$
- Erwartete Ausgabe
 - $A < B$, also $Y = 32'b1$
- Steuereingang für SLT: $F_{2:0} = 3'b111$
- $F_2 = 1'b1$ konfiguriert Addierer als Subtrahierer
 - $S = 25 - 32 = -7$
 - Im Zweierkomplement
 $-7 = 32'h0xffffffff9 \rightarrow \text{msb } S_{31} = 1$
- $F_{1:0} = 2'b11$ wählt $Y = S_{31}$ als Ausgabe
- $Y = S_{31}$ (zero extended) = $32'h00000001$.

Schiebeoperationen (*shifter*)

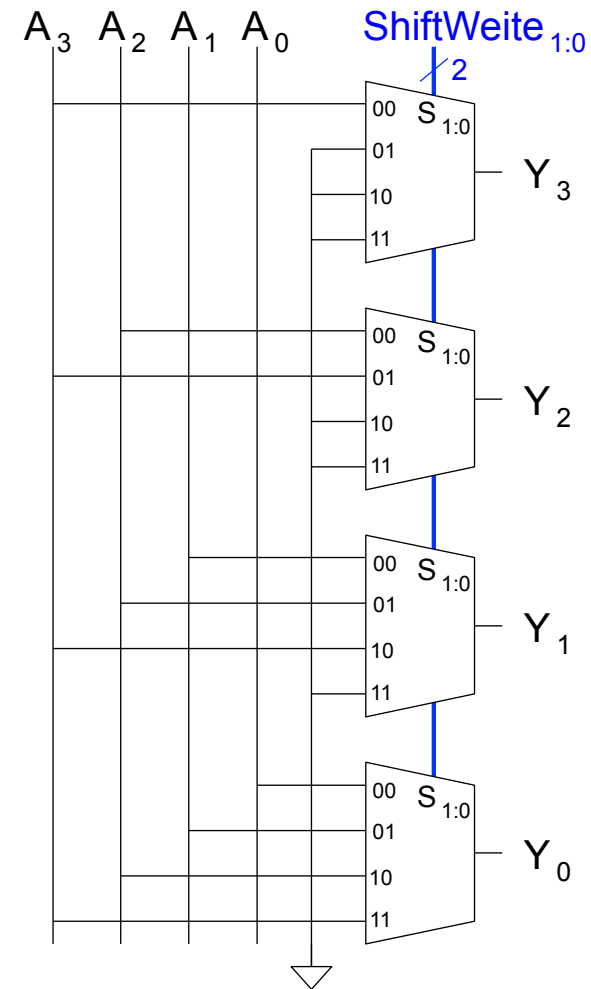
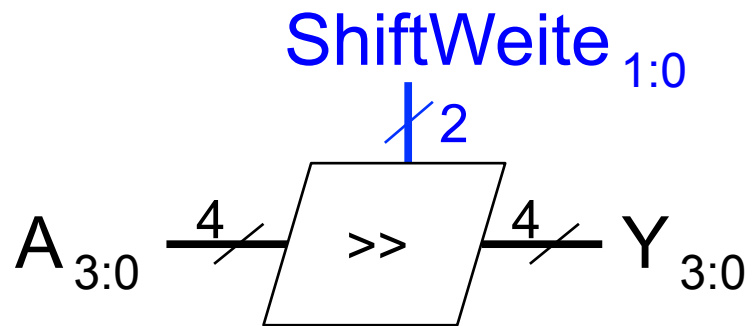
- **Logisches Schieben:** Wert wird eine Bitposition verschoben, leere Stellen mit 0 aufgefüllt
 - Beispiel: $11001 \gg 2 =$
 - Beispiel: $11001 \ll 2 =$
- **Arithmetisches Schieben:** wie logisches Schieben. Verwende aber beim Rechtsschieben alten Wert des msb zum Auffüllen leerer Stellen
 - Beispiel: $11001 \ggg 2 =$
 - Beispiel: $11001 \lll 2 =$
- **Rotierer:** rotiert Bits im Kreis, herausgeschobene Bits tauchen am anderen Ende wieder auf
 - Beispiel : $11001 \text{ ROR } 2 =$
 - Beispiel : $11001 \text{ ROL } 2 =$

LEHRE WIKI
TEST IN
DREI
FOLIEN

Schiebeoperationen (*shifter*)

- **Logisches Schieben:** Wert wird eine Bitposition verschoben, leere Stellen mit 0 aufgefüllt
 - Beispiel: $11001 \gg 2 = 00110$
 - Beispiel: $11001 \ll 2 = 00100$
- **Arithmetisches Schieben:** wie logisches Schieben. Verwende aber beim Rechtsschieben alten Wert des msb zum Auffüllen leerer Stellen
 - Beispiel: $11001 \ggg 2 = 11110$
 - Beispiel: $11001 \lll 2 = 00100$
- **Rotierer:** rotiert Bits im Kreis, herausgeschobene Bits tauchen am anderen Ende wieder auf
 - Beispiel : $11001 \text{ ROR } 2 = 01110$
 - Beispiel : $11001 \text{ ROL } 2 = 00111$

Aufbau von Shiftern



- Bitte jetzt auf LEHRE WIKI eine Frage beantworten!

Shifter als Multiplizierer und Dividierer

- Logisches Schieben um N Stellen nach links multipliziert den Zahlenwert mit 2^N
 - Beispiel : $00001 \ll 2 = 00100$ ($1 \times 2^2 = 4$)
 - Beispiel : $11101 \ll 2 = 10100$ ($-3 \times 2^2 = -12$)
- Arithmetisches Schieben um N Stellen nach rechts dividiert den Zahlenwert durch 2^N
 - Beispiel : $01000 \ggg 2 = 00010$ ($8 \div 2^2 = 2$)
 - Beispiel : $10000 \ggg 2 = 11100$ ($-16 \div 2^2 = -4$)

Multiplizierer

- Schrittweise Multiplikation in Dezimal- und Binärdarstellung:
 - Multiplizieren des Multiplikanden mit einzelner Stelle des Multiplikators
 - Berechnet ein Teilprodukt (auch partielles Produkt genannt)
 - Entsprechend der Wertigkeit der aktuellen Multiplikatorstelle nach links verschobene partielle Produkte werden aufaddiert

Dezimal

$$\begin{array}{r} 230 \\ \times 42 \\ \hline 460 \\ + 920 \\ \hline 9660 \end{array}$$

$$230 \times 42 = 9660$$

Multiplikand
Multiplikator
partielle
Produkte

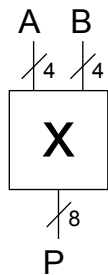
Ergebnis

Binär

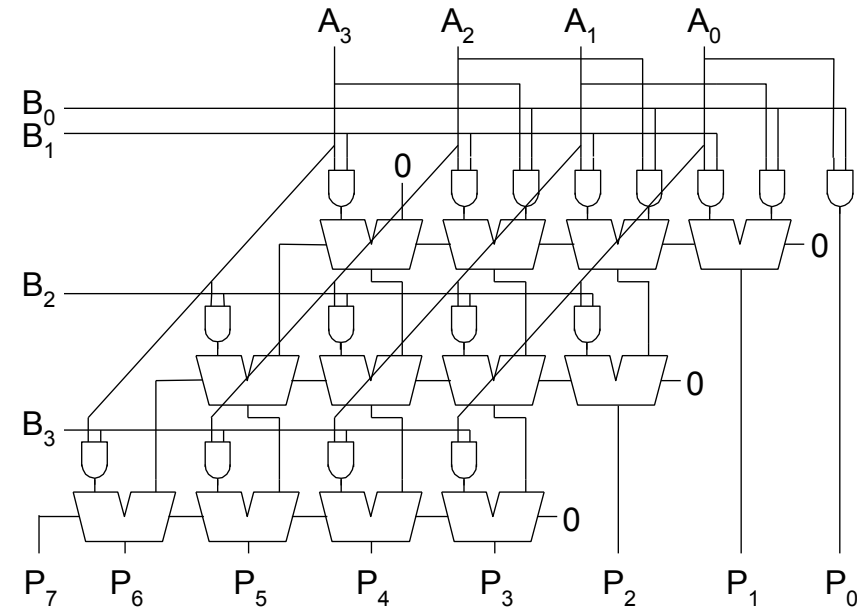
$$\begin{array}{r} 0101 \\ \times 0111 \\ \hline 0101 \\ 0101 \\ 0101 \\ + 0000 \\ \hline 0100011 \end{array}$$

$$5 \times 7 = 35$$

4 x 4 Multiplizierer



$$\begin{array}{r}
 \begin{array}{cccc}
 & A_3 & A_2 & A_1 & A_0 \\
 \times & B_3 & B_2 & B_1 & B_0 \\
 \hline
 & A_3B_0 & A_2B_0 & A_1B_0 & A_0B_0 \\
 & A_3B_1 & A_2B_1 & A_1B_1 & A_0B_1 \\
 & A_3B_2 & A_2B_2 & A_1B_2 & A_0B_2 \\
 + & A_3B_3 & A_2B_3 & A_1B_3 & A_0B_3 \\
 \hline
 P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array}
 \end{array}$$



Multiplikation von k -bit Zahlen hat $2k$ -bit breites Produkt