

SPEZIFIKATION

Flexible ALU

Version	Bearbeiter	Neuerung
1.0		

Inhaltsverzeichnis

1	Preamble	2
2	Datenblatt	3
2.1	Einsatzbereich	3
2.2	Features	3
2.3	Funktion	3
2.4	Blockschaltbild	4
2.5	Schnittstellen	4
2.6	Gate-Count	6
2.7	Timing-Diagramm	8
2.8	Leistungsabschätzung	8
3	Spezifikation	9
3.1	Register	9
3.1.1	RegA	9
3.1.2	RegB	9
3.1.3	AccuA	9
3.1.4	AccuB	9
3.1.5	CMD	9
3.1.6	Flags	9
3.2	Modul-Beschreibung	9
3.2.1	74181	9
3.2.2	MUL-4	9
3.2.3	Barrel-Shifter-8	10
3.2.4	Bi-Reg-n	10
3.2.5	Uni-Reg-n	10
3.2.6	MUX-nxm	10
3.2.7	MC-PROM	10
3.3	Detailblockschaltbild	10

1. *Preamble*

2. Datenblatt

2.1 Einsatzbereich

Diese flexible ALU kann als arithmetisch/logische Einheit in einem Mikroprozessor verwendet werden. **Durch zwei in Microcode realisierbare Befehle...**

2.2 Features

- Addition und Subtraktion zweier 4 Bit Operanden erzeugen ein 5 Bit langes Ergebnis.
- Multiplikation zweier 4 Bit Operanden erzeugt ein 8 Bit langes Ergebnis.
- Native Befehle werden in 2 Takten verarbeitet (Microcode ausgeschlossen).
- **2 Akkumulator Register für Microcode.**
- Für Microcode kann ein Barrel-Shifter verwendet werden.
- Alle Ein- und Ausgänge werden in Registern erhalten.

2.3 Funktion

CMD			Arithmetisch/logischer Ausdruck
2	1	0	
0	0	0	AND
0	0	1	OR
0	1	0	NOT
0	1	1	ADD
1	0	0	SUB
1	0	1	MUL
1	1	0	MCo
1	1	1	MC1

Tabelle 2.1: Befehls-Codierung

Für die Liste aller Port siehe Seite 5 Abbildung 2.2.

- Der Befehl AND legt das Ergebnis an den ersten Operanden A0..3 an.
- Der Befehl OR legt das Ergebnis an den ersten Operanden A0..3 an.
- Der Befehl NOT negiert die Bits der Ports A0..3 und legt das Ergebnis an diesen wieder an.

- Der Befehl ADD addiert Ao..3 und Bo..3 und legt das Ergebnis an Ao..3 an. Tritt bei der Berechnung ein Carry auf, wird dieser in Bo dargestellt und das Ergebnis ist somit 5 Bit lang.
- Der Befehl SUB subtrahiert Bo..3 von Ao..3 und gibt das Ergebnis an Ao..3 aus. Dieses Ergebnis befindet sich im 2er-Komplement!
- Der Befehl MUL multipliziert Ao..3 und Bo..3 und legt das 8 Bit breite Ergebnis an diese Ports an.
- Bei den Befehlen MCo und MC1 handelt es sich zum Zeitpunkt der Anfertigung des Datenblattes um keine konkreten Befehle. Diese werden mit Hilfe von Microcode implementiert. Eine Erklärung dazu erhalten Sie im Spezifikationsteil dieses Dokumentes in Kapitel 3.2.7.

2.4 Blockschaltbild

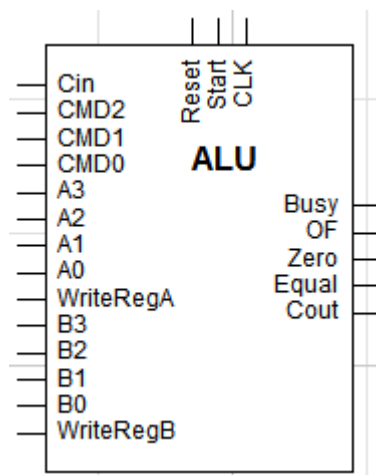


Abbildung 2.1: Äußeres Blockschaltbild

2.5 Schnittstellen

Für detaillierte Informationen zu Ein- und Ausgängen siehe Tabelle 2.2.

Port	Typ	Funktion
Cin	Input	Carry-In für ADD und SUB
WriteRegA	Input	Wenn 1, wird Operand 1 in Register A geschrieben (solange die Arbitrierung außen liegt).
WriteRegB	Input	Wenn 1, wird Operand 2 in Register B geschrieben (solange die Arbitrierung außen liegt).
Reset	Input	Register werden asynchron (ohne Taktflanke) zurückgesetzt.
Start	Input	Die Operation wird mit den angelegten Operanden durchgeführt.
CLK	Input	Takt.
CMD0..2	Input	Auswahl des Befehls. Codierung siehe S. 3
A0..3	Bidirektional	IN: Operand 1 wird mit WriteRegA angelegt. OUT: Ist Busy 0, liegt hier das Ergebnis der Berechnung an. Die Subtraktion liefert das 2er-Komplement. Multiplikation liefert hier das low Nibble des Ergebnisses.
B0..3	Bidirektional	IN: Operand 2 wird mit WriteRegB angelegt. OUT: Wird Busy 0, liegt hier 0 an, die Multiplikation liefert hier das high Nibble des Ergebnisses.
Busy	Output	Ist 1, solange die ALU arbeitet. Wird Busy 0, liegt das Ergebnis an A0..3 und B0..3 an. Die Arbitrierung von RegA und RegB liegt außen, solange Busy=0 ist.
OF	Output	OF ist 1, wenn die letzten beiden Übertragsbits ungleich sind und somit ein Overflow auftritt.
Zero	Output	Zero ist 1, wenn alle Ergebnisbits 0 sind.
Equal	Output	Equal ist 1, wenn die Operanden A und B bitweise gleich sind.
Cout	Output	Carry out. Tritt während der Berechnung von ADD oder SUB ein Übertrag über die letzte Bitstelle auf, wird Cout 1 und kann weiter behandelt werden.

Tabelle 2.2: Liste aller Ports

Tabelle 2.3 können Sie Informationen zur Relevanz der ausgegebenen Flags entnehmen.

Flag	relevant für	Auswirkung
Cout	Addition, Subtraktion	Das Bit ist Repräsentant für einen in das nächste Nibble übergehenden Carry und muss zur Ermittlung des Ergebnisses berücksichtigt werden.
OF	Subtraktion	Die Subtraktion liefert $OF = 1$, wenn ein Carry auftritt und die MSBits der Operanden im Zweierkomplement gleich sind.
Equal	beeinflusst kein Ergebnis	
Zero	beeinflusst kein Ergebnis	

Tabelle 2.3: Flag-Relevanz

2.6 Gate-Count

Für eine detaillierte Angabe der Gatteräquivalente siehe Abbildung 2.4 auf Seite 7.

Typ	Gatteräquivalente	Anzahl	Summe
Gatter			
AND2	1,5	65	97,5
AND3	2	13	26
AND4	2,5	12	30
AND5	3	4	12
Inv	0,5	11	5,5
Mult2:1	3	28	84
Mult4:1	7	0	0
Mult8:1	16	9	144
NAND2	1	6	6
NAND3	1,5	0	0
NAND4	2	0	0
NAND6	4,5	0	0
NAND8	5,5	0	0
NOR2	1	0	0
NOR3	1,5	0	0
NOR4	2	0	0
OR2	1,5	21	31,5
OR3	2	4	8
OR4	2,5	9	22,5
OR5	3	4	12
XNOR2	3	0	0
XOR2	3	32	96
Speicher (Angabe / bit)			
DFF	7	4	28
DFF-R	8	0	0
DFF-S	8	0	0
DFF-SR	9	0	0
DRAM	5	0	0
DRAM (o. Anst.)	0,25	0	0
SRAM	7,5	0	0
SRAM (o. Anst.)	1	0	0
Buffer4	4	19	76
74_181	108,5	1	108,5
Gesamtsumme:			787,5

Tabelle 2.4: Gatteräquivalente

2.7 Timing-Diagramm

2.8 Leistungsabschätzung

watt pro gatter pro mhz als energieverbrauch

3. Spezifikation

3.1 Register

3.1.1 RegA

Register für Operand A und das low-nibble des Ergebnisses. Es handelt sich um ein Bi-Reg-4. Die Arbitrierung des Registers liegt intern, solange busy=0 ist. Das Ergebnis liegt am Ausgang an, sobald busy=0 wird.

3.1.2 RegB

Register für Operand B und das high-nibble des Ergebnisses. Es handelt sich um ein Bi-Reg-4. Die Arbitrierung des Registers liegt intern, solange busy=0 ist. Das Ergebnis liegt am Ausgang an, sobald busy=0 wird.

3.1.3 AccuA

Akkumulatorregister für das low-nibble des Ergebnisses. Es handelt sich um ein Uni-Reg-4. Die Ausgänge des Registers können als Operand A verwendet werden.

3.1.4 AccuB

Akkumulatorregister für das high-nibble des Ergebnisses. Es handelt sich um ein Uni-Reg-4. Die Ausgänge des Registers können als Operand B verwendet werden.

3.1.5 CMD

Register für den Befehlscode. Es handelt sich um ein Uni-Reg-3. Wird geschrieben sobald Start=1 gesetzt wird.

3.1.6 Flags

Register für die Flags. Es handelt sich um ein Uni-Reg-4. Die Flags liegen am Ausgang an, sobald busy=0 wird.

3.2 Modul-Beschreibung

3.2.1 74181

Die 74181 dient zur Berechnung aller nativen ALU-Befehle außer der Multiplikation.

3.2.2 MUL-4

Ein 4x4 Bit Hardware-Multiplizierer mit 8 Bit Ausgang.

3.2.3 Barrel-Shifter-8

Ein 8-Bit Barrel-Shifter, der links/rechts rotieren/schieben um 0-7 bit kann.

3.2.4 Bi-Reg-n

3.2.5 Uni-Reg-n

Ein

3.2.6 MUX-nxm

Multiplexer zum Schalten von n Signalquellen mit je m Bits.

3.2.7 MC-PROM

Programmierung

3.3 Detailblockschaltbild

Für das innere Schaltbild siehe Seite 10.

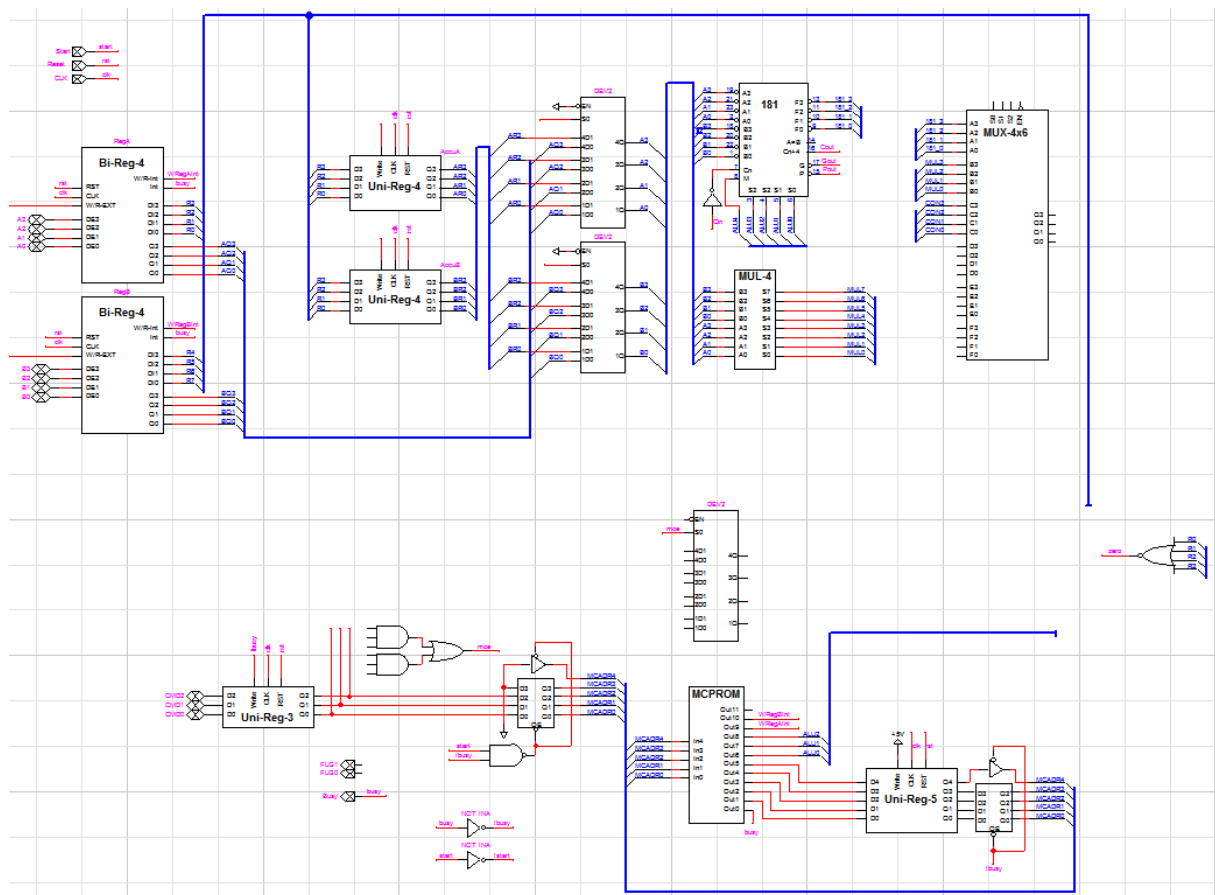


Abbildung 3.1: Inneres Blockschaltbild