

Aufgabe 1.1: Technologien und Grundlagen (12,5 Punkte)

a) Gegeben sei folgende Wahrheitstabelle:

a	b	c	$f(a,b,c)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Geben Sie $f(a, b, c)$ in disjunktiver kanonischer Normalform (DKN) an.

b) Vereinfachen Sie $f(a, b, c)$ algebraisch soweit wie möglich.

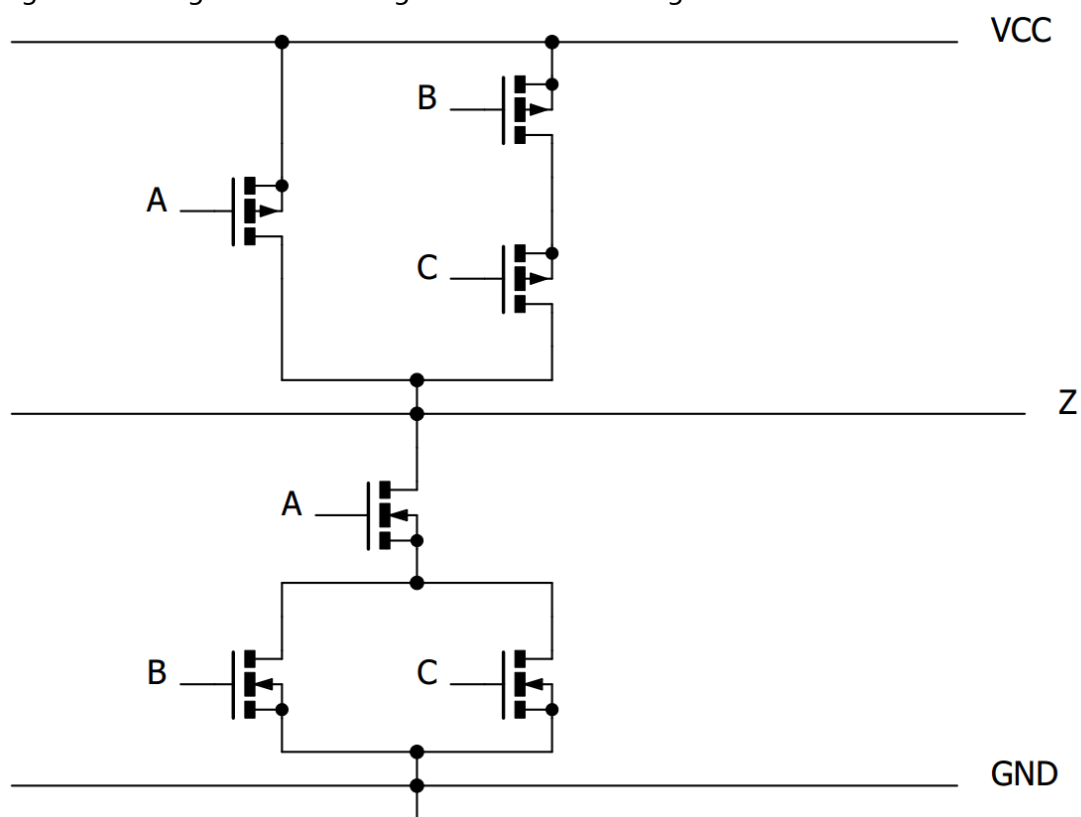
- c) Formen Sie $f(a, b, c)$ so um, dass sich die Schaltfunktion ausschließlich aus NAND-Ausdrücken über jeweils zwei Termen zusammensetzt. Ihnen stehen keine Inverter zur Verfügung. Verwenden Sie für die finale Darstellung die Sheffer-Notation $(a | b)$.

- d) Zeichnen Sie den Schaltplan der Schaltungsfunktion $f(a, b, c)$ unter Verwendung von Schaltsymbolen neuer DIN-Norm bestehend aus NAND-Gattern mit zwei Eingängen.

- e) Zeichnen sie ein RS-Flipflop mit Schaltsymbolen neuer DIN-Norm. Wegen Lieferengpässen stehen Ihnen leider nur noch ein NAND-Gatter, ein NOR-Gatter und vier Inverter zur Verfügung.



- f) Gegeben sei folgende Schaltung in CMOS-Technologie.



Bestimmen Sie $f(A,B,C) = Z$.

- g) Zeichnen Sie eine Schaltung in RTL-Technologie, welche die Schaltfunktion $f(A,B,C)$ aus f) realisiert. Verwenden Sie ausschließlich *npn*-Transistoren und Widerstände. Sollten Sie f) nicht gelöst haben, verwenden Sie folgende Schaltfunktion:

$$(A \downarrow C) \mid (A \downarrow B)$$

Aufgabe 1.2: Steuerwerksentwurf

(12,5 Punkte)

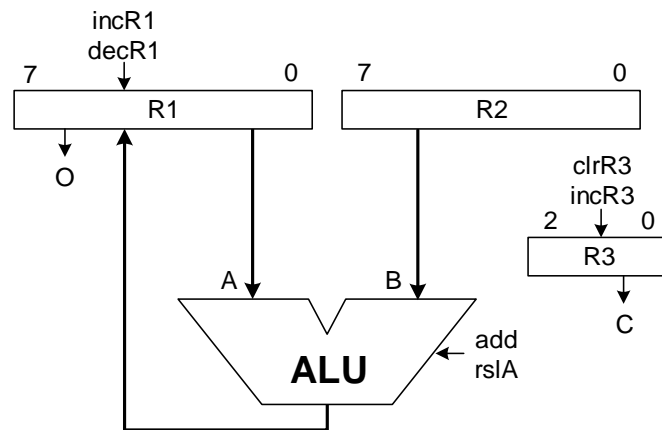


Abbildung 1: Operationswerk

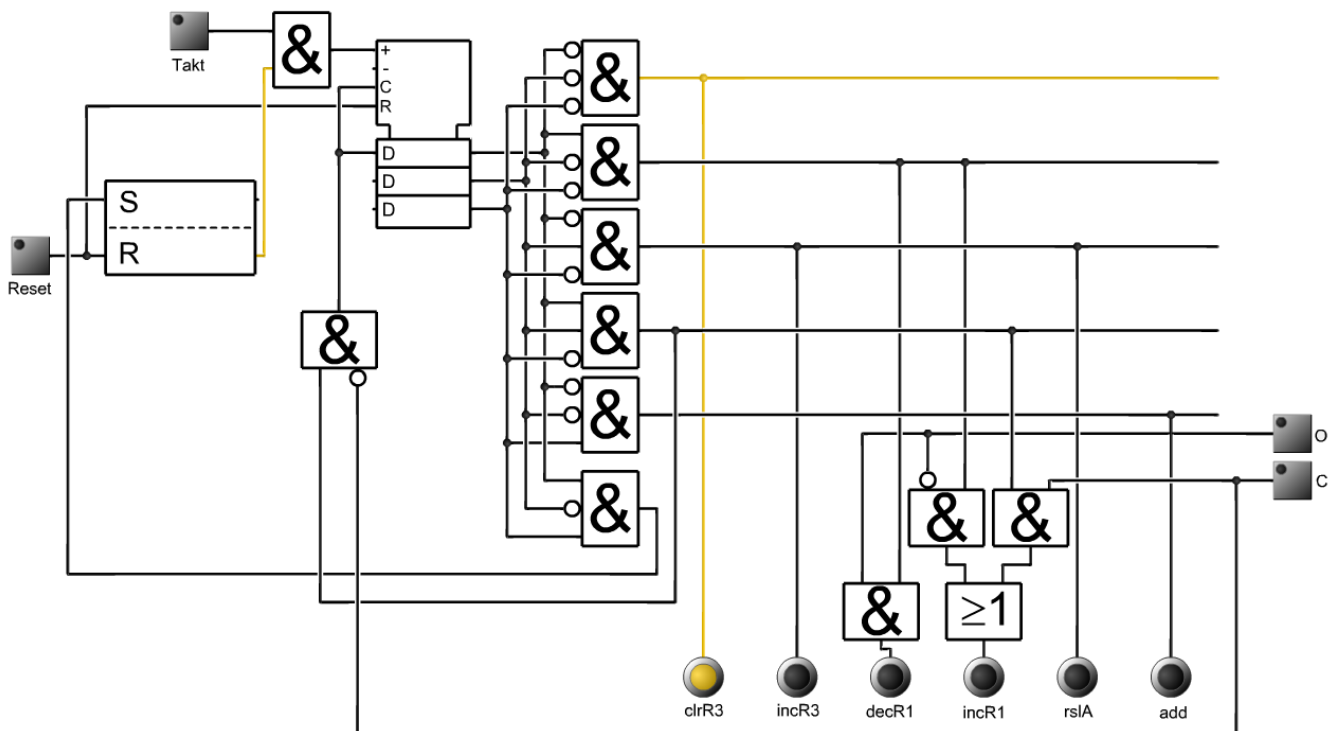


Abbildung 2: Steuerwerk

Das gegebene Operationswerk aus Abbildung 1 basiert auf den zwei 8-Bit Registern R_1 und R_2 , dem 3-Bit Register R_3 und einer 8-Bit ALU. Sowohl die Eingänge als auch der Ausgang der ALU und Register sind 8 Bit breit. Mithilfe zweier Steuersignale können Operationen auf der ALU ausgeführt werden. Das Steuersignal *add* addiert die beiden an den Eingängen anliegenden Wert und das Signal *rslA* führt einen Ringshift auf dem am Eingang A anliegenden Wert um eine Stelle nach links durch. Das Register R_1 kann mithilfe des Steuersignals *incR1* inkrementiert und mithilfe des Signals *decR1*

dekrementiert werden. Das Register R_3 lässt sich mit dem Steuersignal $clrR_3$ auf den Wert Null zurücksetzen und mit dem Signal $incR_3$ inkrementieren. Es setzt zudem für einen Takt das Kriterium C auf 1, sollte durch das Inkrementieren ein Überlauf im Register stattfinden. Das Register R_1 führt das Kriterium O (odd) nach außen, welches gesetzt wird, sollte der Zahlenwert in R_1 ungerade sein.

- a) Um welche Art von Steuerwerk handelt es sich?

- b) Analysieren Sie das in Abbildung 2 gegebene Steuerwerk für das Operationswerk aus Abbildung 1, indem Sie die nachfolgende Tabelle 1 um die Steuersignale ergänzen.
- c) Geben Sie nun die zu den Steuersignalen korrespondierenden Registertransferoperationen an, indem Sie den RT-Quellcode in Tabelle 1 ergänzen.
- d) Welche der nachfolgend genannten Berechnungen werden teilweise oder als Ganzes vom Steuerwerk auf dem Operationswerk ausgeführt? Bitte kreuzen Sie die zutreffenden Antworten an. Flasche Antworten führen zu Punktabzügen. Es kann jedoch kein negativer Punktwert für diese Teilaufgabe erzielt werden.
- ☐ Berechnung der Hamming-Distanz von R_1 und R_2
 - ☐ Subtraktion $B-A$
 - ☐ Ringshift von R_2
 - ☐ Ringshift von R_1
 - ☐ Bilden des Einerkomplements von R_1
 - ☐ Bilden des Zweierkomplements von R_1
 - ☐ Berechnung der Summe der in R_1 vorkommenden 1en
- e) Wie oft wird die im Algorithmus vorkommende Schleife (Loop) je Durchlauf des Algorithmus abgearbeitet?

- f) Läuft der durch das Steuerwerk gegebene Algorithmus in einer Endlosschleife?

Tabelle 1

RT-Code	Steuersignale
START: _____	_____
LOOP: if _____ then	
_____	_____
else	
_____	_____
fi;	
_____	_____
_____	_____
_____	_____
if _____ then	
_____	_____
else	
_____	_____
fi;	
_____	_____

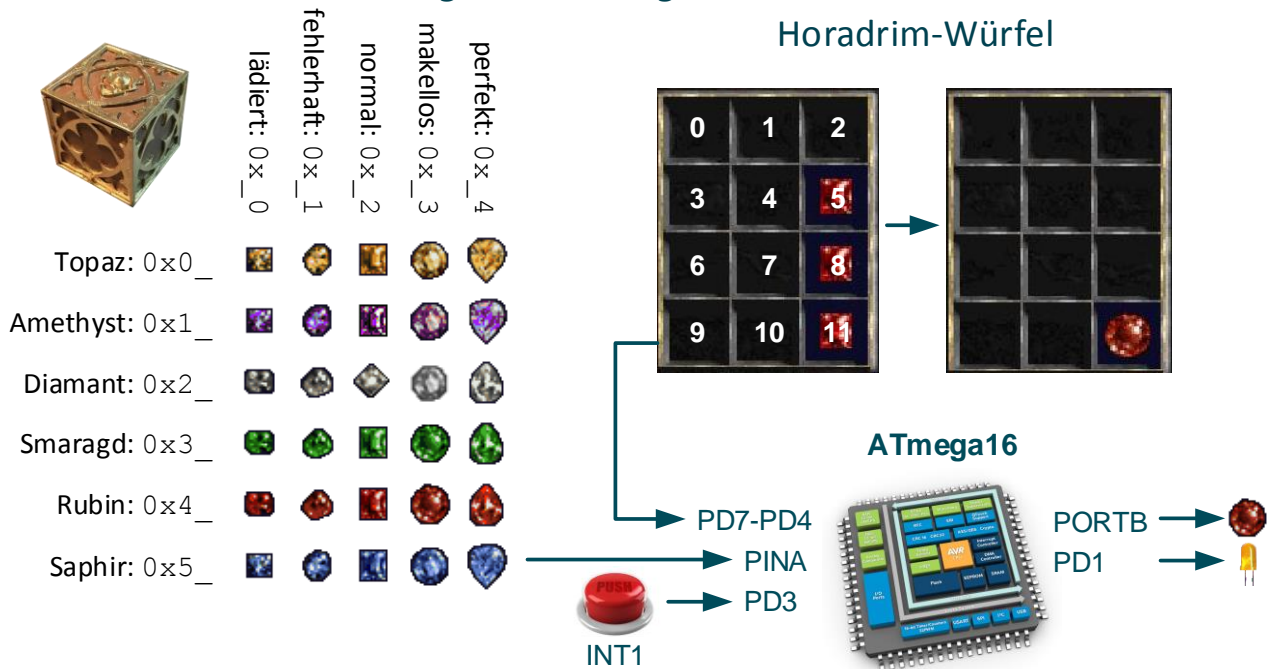
Matrikelnummer: _____

Studiengang: _____

$\Sigma_{A1} =$ _____ Punkte

Aufgabe 2: Assemblerprogrammierung (25 Punkte)

Programmierung des Horadrim-Würfels aus Diablo 2 ©



Im Jahr 2000 erschien Blizzard Entertainments Action-Rollenspiel Diablo 2, welches als wesentlichen Bestandteil der Ausrüstung des Spielers einen sogenannten Horadrim Würfel enthält. Mit diesem können nicht nur Gegenstände verzaubert und aufgewertet werden, sondern auch gefundene Edelsteine niedriger Qualitätsstufen zu Steinen höherer Qualität kombiniert werden. Es gibt im Spiel sieben verschiedene Arten von speziellen Edelsteinen, welche in fünf Qualitätsstufen vorliegen können (niedrigste Qualität = lädiert, höchste Qualität = perfekt). Mit Hilfe des Würfels ist es nun möglich, aus drei Edelsteinen derselben Qualitätsstufe einen Edelstein höherer Stufe zu erhalten. Im obigen Beispiel ergibt sich aus drei normalen Rubinen ein makelloser Rubin.

Ziel dieser Aufgabe ist die teilweise Nachempfindung des Horadrim-Würfels mit Hilfe eines ATmega16 Mikrokontrollers in etwas abgeänderter Form. Zu diesem Zweck wurden die verschiedenen Edelsteine entsprechend der oben gegebenen Tabelle hexadezimal kodiert. Beispielsweise ist ein makelloser Rubin durch 0x43 kodiert. Der Würfel besitzt zwölf Felder, auf denen die Edelsteine beliebig platziert werden können. Ist ein Feld bereits bestzt, so wird der entsprechende Inhalt bei einem erneuten hineinlegen eines Steins überschrieben. Die Nummerierung der einzelnen Felder ist in der obigen Abbildung gegeben. Der Würfel prüft durchgängig, ob er drei Edelsteine desselben Typs und von selber Qualität enthält, die noch zu einem Edelstein einer höheren Qualitätsstufe kombinierbar sind. Ist dies der Fall, so werden die Edelsteine zu einem verschmolzen, der auf dem Feld 11 platziert wird. Der restliche Würfel wird geleert. Der Code des neuen Edelsteins wird anschließend auf PORTB für drei Sekunden lang ausgegeben. Für denselben Zeitraum soll die gelbe LED an PD1 leuchten. Das Betätigen des Tasters an PD3 soll durch das Auslösen eines Interrupts auf der Fallenden Flanke das Hineinlegen eines Edelsteins in den Würfel ermöglichen. Der Code des Edelsteins muss entsprechend an PINA und die Nummer des Feldes an PD7-PD4 angelegt werden. Liegt zum Zeitpunkt des Tastendrucks der Code 0xFF an PINA an, wird der Würfel geleert.

Aufgaben:

Die Bearbeitung der Aufgabe erfolgt in mehreren Unteraufgaben. Nutzen Sie die einzeln vorgegebenen Bereiche und schreiben Sie aussagekräftigen Assembler Code.

- a. Beginnen Sie die Programmierung mit den einzelnen Registernamensdefinitionen. Sorgen Sie dafür, dass das Register R20 unter dem Namen `time` angesprochen werden kann. Reservieren Sie weiterhin im RAM unter dem Namen `cube` zwölf Bytes für den Inhalt des Würfels.

- b. Initialisieren Sie die Interrupt-Vektor-Tabelle, sodass nach einem RESET zum Label `init`, nach einem nach einem *External Interrupt Request 1* zur `INT1_ISR` und nach einem *Timer/Counter 1 Compare Match* Interrupt zur `TIMER1_ISR` gesprungen wird.

- c. Beginnen Sie nun unter dem Label `init` die Programmierung mit der Initialisierung des Stackpointers am Ende des RAMs und konfigurieren Sie die verwendeten Ein-/Ausgabeports.

- d. Konfigurieren Sie den Timer/Counter1 so, dass er beim Erreichen des Vergleichswerts 97 einen Interrupt auslöst. Verwenden Sie einen Prescaler von 1024 und aktivieren Sie den CTC Modus.

- e. Initialisieren Sie im Folgenden den externen Interrupt **INT1**, welcher ausgelöst werden soll, sobald eine fallende Flanke registriert wird. Aktivieren Sie dann die Interrupts global und springen sie anschließend zu **main**.

- f. Implementieren Sie die **TIMER1_ISR** Interrupt Service Routine welche das interne Register **time** inkrementiert.

- g. Schreiben Sie das Unterprogramm `clear_cube`, welches die Startadresse des Würfels im X Register erwartet und alle Felder des Würfels mit dem Wert `0xFF` befüllt.

- h. Implementieren Sie die `INT1_ISR` Interrupt Service Routine, welche den aktuellen an PINA in kodierter Form anliegenden Edelstein in das Feld entsprechende Feld hineinlegt oder gegebenenfalls den Würfel leert, indem jedes Feld den Kode `0xFF` erhält. Sie könne davon ausgehen, dass nur erlaubte Kodes anliegen.

- i. Implementieren Sie nun die in einer Endlosschleife laufenden Grundfunktionalität unter dem Label main. Sie können davon ausgehen, dass nur korrekte Codes in den Feldern stehen und leere Felder mit 0xFF belegt sind.

Ihnen ist ein Unterprogramm check4gems gegeben, welches im X Register die Startadresse der Felder des Würfels erwartet und diesen nach drei gleichen Edelsteinen durchsucht. Der Code der Edelsteine wird in R16 geschrieben. Sollten keine drei gleichen Edelsteine oder mehrere Trios von Edelsteinen vorkommen, liefert es in R16 den Wert 0xFF zurück.

Verwenden Sie für das Warten der drei Sekunden nach einer Erfolgreichen Verschmelzung Polling auf das Register time.

$\Sigma_{A2} = \underline{\hspace{2cm}}$ Punkte