

INSTITUT FÜR MATHEMATISCHE MASCHINEN  
 UND DATENVERARBEITUNG  
 Lehrstuhl für Rechnerstrukturen (IMMD III)  
 Universität Erlangen-Nürnberg  
 Martensstr. 3, 91058 Erlangen

02.08. 1999

Klausur  
 zu

**"Organisation und Technologie von Rechensystemen 4"**

.....                      .....                      .....  
 Matrikelnummer            Geb.-Datum            Vorname                      Name

- Außer Taschenrechner sind keine Hilfsmittel erlaubt.
- Legen Sie den Ausweis (mit Lichtbild!) griffbereit auf den Platz.
- Dieses Aufgabenheft umfaßt 12 Seiten. Überprüfen Sie die Vollständigkeit.
- Gesondert beigelegte Blätter werden nicht bewertet!
- Schreiben Sie deutlich! Unleserliches wird nicht bewertet!
- Es darf nicht mit der Farbe rot geschrieben werden!
- Schreiben Sie die Antworten jeweils in den freien Raum hinter den Fragen; sollte der Platz nicht ausreichen, bitten Sie die Aufsicht um weitere Formblätter. Hier können auch Nebenrechnungen o. ä. stehen.
- Bekanntgabe der Ergebnisse: Aushang in den Semesterferien
- Einsichtnahme: Siehe Aushang
- Bei Bestehen wird der Schein direkt an das Prüfungsamt geschickt!

Durch meine Unterschrift bestätige ich

- den Empfang der vollständigen Klausurunterlagen
- die Kenntnisnahme der obigen Informationen.

Ich bin damit einverstanden, daß mein Prüfungsergebnis der Klausur zu OTR 4 unter Angabe der Matrikel-Nummer veröffentlicht wird.

Erlangen, den 02.08. 1999.....

(Unterschrift)

Aufgabe	1	2	3	4	5	6	Summe
max. Punktzahl	7	5	4	6	7	7	36
erreichte Punktzahl							

**Aufgabe 1: (MC68HC05-Assembler)****(7 Punkte)**

Dem folgenden Unterprogramm wird eine 16-Bit Zahl als Parameter übergeben. Es liefert ein 8-Bit Resultat zurück.

```

was_tut_es: ;subroutine
            ;parameter Zahl : word (stored in zahlHi : zahlLo)
            ;returns   Result: byte (stored in result)
            ;uses A(accu), X(indexreg), tmp1, tmp0

loop:      ldX #0
            stX tmp0
            incX
            clr tmp1
            lsl tmp0           ;left shift (MSB to carry-flag)
            rol tmp1           ;rotate left through carry-flag
            ldA zahlLo
            sec                 ;set carry-flag
            sbc tmp0           ;subtract tmp0 and carry-bit from accu
            stA zahlLo
            ldA zahlHi
            sbc tmp1
            stA zahlHi
            blo m1             ;branch if (A) < (tmp1)
            bne loop
            tst zahlLo         ;test for negative or zero
            bne loop         ;branch if zahlLo <> 0
m0:        stX result
            rts
m1:        decX
            bra m0             ;branch always

```

1a) Beschreiben Sie kurz den dem Programm zugrundeliegenden Algorithmus!

1b) Was ist das Resultat bei Aufruf des Unterprogramms mit den Werten:

24:

25:

26:

1c) Durch das Unterprogramm wird der ganzzahlige Anteil einer arithmetischen Standardfunktion berechnet. Welche Standardfunktion ist das?

**Aufgabe 2: (Interrupts, Ein-/Ausgabe und Register)**

**(5 Punkte)**

2a) Nennen Sie den Hauptvorteil des Interruptkonzepts!

2b) Erläutern Sie das Konzept der Memory Mapped IO!

2c) Nennen Sie 2 Spezialregister (keine Register für Operanden) des Microcontrollers MC68HC05 und erläutern Sie deren Funktion!

**Aufgabe 3: (Adressierung)**

**(4 Punkte)**

Nutzen Sie für die Beispiele Assembler-Instruktionen des Microcontrollers MC68HC05.

3a) Erläutern Sie die folgenden Adressierungsarten und geben Sie jeweils ein Beispiel!

i) Immediate

ii) Direkt

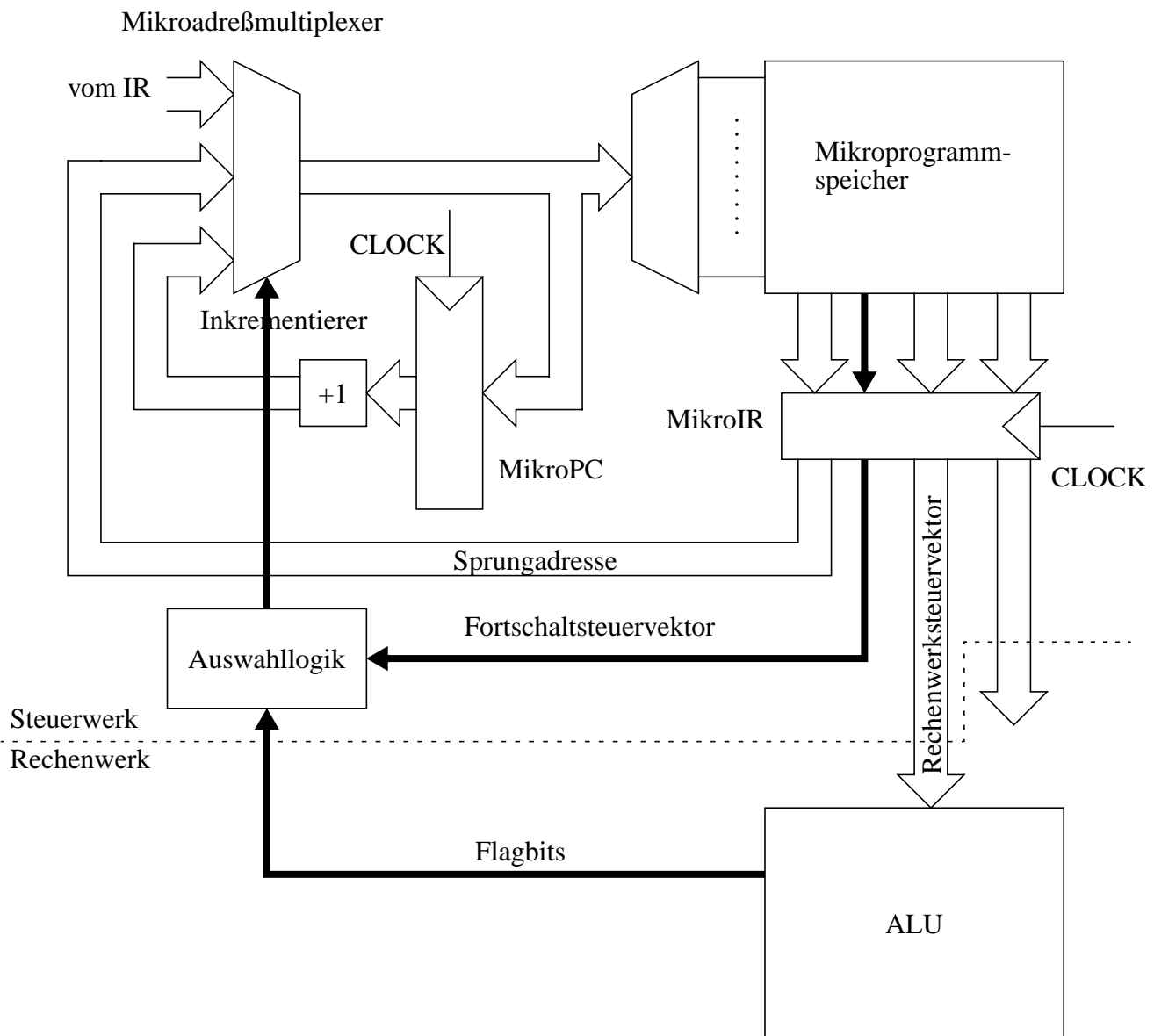
iii) Relativ

3b) Welche Adressierungsart muß ein Prozessor unbedingt beherrschen, um Operanden zu laden, deren Adressen erst zur Laufzeit berechnet werden? Geben Sie ein Beispiel an!

**Aufgabe 4: (Mikroprogrammsteuerung)****(6 Punkte)**

Ein mikroprogrammiertes Steuerwerk mit dem im Bild angegebenen Aufbau hat folgende maximale Verzögerungszeiten für die einzelnen Elemente:

Register	von der positiven Taktflanke (CLOCK) bis zum Gültigwerden der Ausgänge	2ns
ALU	vom Anlegen des Steuercodes bis zum Gültigwerden der Flags	17 ns
Auswahllogik		4 ns
Multiplexer		4 ns
Inkrementierer		10 ns



- 4a) Das Steuerwerk soll mit einer Taktfrequenz von 20 MHz betrieben werden. Wie hoch darf die Zugriffszeit für den Mikroprogrammspeicher höchstens sein? Geben Sie die Herleitung an!
- 4b) Erläutern Sie den Unterschied zwischen horizontalem und vertikalem Mikrobefehlsformat und zählen Sie die jeweiligen Vor- und Nachteile auf!
- 4c) Erläutern Sie den Begriff der Emulation im Bereich der Mikroprogrammierung! Nennen Sie zwei Beispiele, wofür Emulation verwendet werden kann!

**Aufgabe 5: (Mikroprogrammierung)****(7 Punkte)**

Ein mikroprogrammiertes Steuerwerk interpretiert jeden Maschinenbefehl durch eine Folge von Mikroschritten, die im zugehörigen Mikroprogramm festgelegt sind.

Für das hier betrachtete Steuerwerk (Fortschaltsteuerung siehe Bild bei Aufgabe 4) gelte:

Zu Beginn jedes Mikroprogramms stehe der Opcode des Maschinenbefehls im Instruktionsregister IR und der Programmcounter PC zeige auf das Folgewort im Maschinenprogramm. Jedes Mikroprogramm terminiert mit einem Einsprung in das Mikroprogramm, das durch den Inhalt von IR spezifiziert ist. Das Speicherinterface bestehe aus dem Speicheradreibregister SAR und dem Speicherdatenregister SDR. SP enthalte den Stackpointer.

Ein Mikrobefehl für einen 1-Bus-Datenpfad hat folgende Steuerfelder:

**BUS:** Der erste Operand bezeichnet die Quelle, die auf den Bus geschaltet wird. Der zweite Operand bezeichnet die Zielregister, die am Zyklusende das Datum vom Bus übernehmen.

**ALU** spezifiziert die Verknüpfung zwischen den ALU-Eingangsregistern A und B. Das Ergebnis wird (außer bei nop) am Zyklusende im Ergebnisregister C gespeichert.

**SPST** steuert die Speicherzugriffe: read lädt am Zyklusende das SDR mit dem durch SAR adressierten Wert; write schreibt den Inhalt von SDR in den Speicher.

**MIR\_D** ist ein Teilfeld des Mikrobefehlsregisters, das als Quelle auf den Bus geschaltet werden kann.

Das folgende Mikroprogramm interpretiert den Maschinenbefehl **jump\_subroutine sprungziel** durch eine Steuersequenz.

BUS		ALU	SPST	Mikrofortschaltung	Sprungadresse bzw. MIR_D
Quelle	Ziele				
SP	A	nop	nop	continue	-
MIR_D	B	nop	nop	continue	2
PC	A, SAR	add	nop	continue	-
C	SP, SAR	nop	read	continue	-
SDR	A, PC	add	nop	continue	-
C	SDR	add	nop	continue	-
PC	SAR	nop	write	continue	-
C	PC	nop	read	continue	-
SDR	IR	nop	nop	continue	-
-	-	nop	nop	jump(IR)	-

5a) Kreuzen Sie diejenige Variante für den push-Vorgang an, welche hier offenbar gewählt wurde!

- post decrement
- post increment
- pre decrement
- pre increment





**Schmierblatt: diese Seite wird nicht bewertet**

**Schmierblatt: diese Seite wird nicht bewertet**