

## M32- Quick-Reference-Card

### Register (alle 32 Bit):

<b>PC</b>	Program-Counter, Befehlszähler	Logische Adresse, von wo die CPU den nächsten Befehl holt
<b>STATUS</b>	Status-Bits	Siehe unten
<b>R0,...,R7</b>	Allgemeine Register	Frei benutzbar
<b>INTFLAGS</b>	Interrupt-Flags	Siehe unten
<b>INTBASE</b>	Basis der Interrupt-Sprungtabelle	
<b>TIMER</b>	Timer Inhalt	Zähler, hier steht der aktuelle Wert
<b>TRELOAD</b>	Timer-Nachladewert	Wert, mit dem der Zähler beim Unterlauf neu gesetzt wird
<b>SP,SSP,USP</b>	Stack-Pointer, Stapelzeiger	SSP,USP: Stack-Pointer, TOS angeben, Stapel wächst nach unten Nach dem Retten eines Wertes auf dem Stapel wird SP erniedrigt
<b>BASE SBASE UBASE</b>	MMU Basisregister des Segments	...wird auf die logische Adresse aufaddiert
<b>LIMIT SLIMIT ULIMIT</b>	MMU Grenzregister des Segments	SLIMIT wird geprüft: $a < \text{SLIMIT} \rightarrow$ physik. Adr. = $a + S - \text{UBASE}$ $a > \text{SLIMIT} \rightarrow$ Speicherzugriffsverletzung SLIMIT=0 : Test ausgeschaltet

<b>Status-Register STATUS:</b>	<b>Interrupt-Register INTFLAG:</b>
Bit 0: ZERO Bit Bit 1: Greater-Equal Bit 8: Timer-Run Bit 16: User Bit 17: MMU-Fault	Bit 0: INTENA Interrupt-enable, allgemeine Interrupt-Freigabe Bit 1: TIENA Timer-Interrupt-Enable Bit 2: TINT Timer-Interrupt-Bit Bit 3: EX1ENA EXTERN1-Interrupt-Enable Bit 4: EX1INT EXTERN1-Interrupt-Bit Bit 5: EX2ENA EXTERN2-Interrupt-Enable Bit 6: EX2INT EXTERN2-Interrupt-Bit (freigeben = 1, sperren = 0)

### Operanden, Adressierungsarten:

Notation	Beispiel	Bezeichnung	Bedeutung	Verwendung
<b>nn</b>	100H	immediate	nn ist selbst der Operand	R
<b>reg</b>	STATUS	register	Der Registerinhalt ist der Operand	R,L
<b>TOS</b>	TOS	top of stack	Das oberste Stackelement ist der Operand	R,L
<b>nn(reg)</b>	17(R7)	reg. relativ	Registerinhalt + nn ist Adresse des Operanden	R,L,A
<b>@nn</b>	@100	direct	nn ist die Adresse des Operanden	R,L,A

### Indizierte Adressierung:

Notation	Beispiel	Bezeichnung
<b>nn(reg1)[reg2]</b>	17(R7)[R2]	reg. relativ, indexed
<b>@nn[reg1]</b>	@100[R3]	direct, indexed

Zur nicht-indizierten Adresse wird dann der Inhalt des spezifizierten (in eckigen Klammern notierten) Registers hinzuaddiert.  
Indizierung ist nur bei Typ-A Operanden zulässig

### M32 Prozessor Befehle:

Befehl	Funktion (JAVA)	Beschreibung
<b>ADD a,b</b>	$a = a + b$	Addition
<b>SUB a,b</b>	$a = a - b$	Subtraktion
<b>AND a,b</b>	$a = a \& b$	bitweises UND
<b>OR a,b</b>	$a = a   b$	bitweises OR
<b>XOR a,b</b>	$a = a \wedge b$	bitweises XOR
<b>NOT a,b</b>	$a = \sim b$	Einerkomplement
<b>MUL a,b</b>	$a = a * b$	Multiplikation
<b>DIV a,b</b>	$a = a / b$	Division
<b>MOD a,b</b>	$a = a \% b$	Modulo-Operation

#### Zuweisungsbefehl:

<b>MOV a,b</b>	a = b	Zuweisung
----------------	-------	-----------

#### Vergleichsbefehl:

<b>CMP a,b</b>	a>=b?	Vergleich der Operanden a und b, Greater-Equal-Bit wird gesetzt, wenn a>=b ist. Das ZERO Bit wird gesetzt, wenn a=b ist.
----------------	-------	--

#### Weitere Befehle:

<b>NOP</b>		Keine Operation
<b>HALT</b>		Halt der CPU

#### Sprungbefehle:

<b>JMP a</b>		Springe zu Adresse a
<b>JNZ a</b>		Springe, wenn ZERO Bit nicht gesetzt zu Adresse a
<b>JZ a</b>		Springe wenn ZERO Bit gesetzt zu Adresse a
<b>JGT a</b>		Springe, wenn Greater-Equal-Bit gesetzt und Zero-Bit nicht gesetzt zu Adresse a
<b>JGE a</b>		Springe, wenn Greater-Equal-Bit gesetzt zu Adresse a

#### Unterprogrammaufrufe:

<b>CALL a</b>		Unterprogrammaufruf zu Adresse a
<b>RET</b>		Rückkehr vom Unterprogramm

#### Interruptbefehle:

<b>INT n</b>	n (0..15)	Führe Softwareinterrupt n aus
<b>RETI</b>		Rückkehr vom Interrupt

#### Ausgabebefehle:

<b>PRN a</b>		hexadezimale Ausgabe des Operanden
<b>PRT a</b>		Textausgabe des bei der Adresse des Operanden stehenden Textes
<b>PRL a</b>		wie PRT, aber mit Zeilenvorschub (Linefeed) nach der Ausgabe

Die Operanden beim PRT und PRL Befehl müssen A-Werte sein, die Text im Speicher adressieren. Textende wird im Speicher durch 0 markiert!

### Die M32 Assemblersprache:

#### Aufbau einer Zeile:

Marke (Label)	Befehl	Operanden	;Kommentar
START:	MOV	R0,15	;das ist der Zuweisungsbefehl

#### Assembleranweisungen:

<b>ORG konstante</b>	Platzzeiger erhält Wert der Konstanten
<b>OFFSET konstante</b>	OFFSET-Wert wird auf den Wert der Konstanten gesetzt
<b>name EQU konstante</b>	In der Symboltabelle wird dem Symbol „name“ der Wert der Konstanten zugewiesen
<b>DS konstante</b>	Platzzeiger wird um den Wert der Konstanten erhöht
<b>DW konstante{,konstante}</b>	An der Stelle des Platzzeigers Konstanten ablegen
<b>INCLUDE „name“</b>	Datei mit dem Namen „name“ einfügen
<b>END</b>	Bezeichnung für das Ende der Assembler-Quelldatei

Erstellt von: Johannes Bloecker, 5.11.2001

Bearbeitet: M. Oßmann 27.9.2002, Für Hinweise auf enthaltene Fehler sind wir dankbar !