Entwickler Dokumentation

Inhalt

(1) Microprogramm *strtolower_[RB]*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
strtolower_[RB]																											
010	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	NOP	7	MR	f	IR	AB	H	CI0	X	L	H	77	PS	CONT	010	H	H	H	H	H	R
011	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	SUBS	QREG	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	000	H	H	H	H	H	R
012	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	f	MR	f	MR	H	H	CI1	X	H	H	25	C	CJP	000	H	H	H	H	H	R
013	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	0041	DQ	SUBR	NOP	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	000	H	H	H	H	H	R
014	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	7	MR	f	MR	H	H	CI0	X	H	H	32	C	CJP	01A	H	H	H	H	H	R
015	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	005A	DQ	SUBR	NOP	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	015	H	H	H	H	H	R
016	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	7	MR	f	MR	H	H	CI0	X	H	H	34	C	CJP	01A	H	H	H	H	H	R
gross->klein																											
017	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	0020	DQ	ADD	QREG	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CONT	017	H	H	H	H	H	R
018	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	NOP	7	MR	f	IR	AB	H	CI0	X	H	H	77	PS	CONT	018	H	H	H	H	H	W
019	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZQ	ADD	NOP	7	MR	f	MR	H	DB	CI0	X	H	H	77	PS	CONT	000	H	H	H	H	H	R
RB+1																											
01a	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	RAMF	7	MR	f	IR	H	H	CI1	X	H	H	77	PS	CJP	010	H	H	H	H	H	R

Strtolower_[RB]	Wandelt alle vorkommenden Kleinbuchstaben eines Wortes ab der Adresse im Register RB in Großbuchstaben um (Ende eines Wortes ist 0x0000 (NULL))
1. Takt: 010	1. RB wird auf den Adressbus gelegt, um das erste Zeichen des Wortes zu erhalten
2. Takt: 011	1. Am K-Mux liegt nun der Wert des ersten Zeichens an, welches nach einer Subtraktion mit 0 in das Q-Register geschrieben wird
3. Takt: 012	1. Durch die Vorherige Subtraktion wurden bereits die Flags für einen Vergleich gleich 0 gesetzt, was nun mit srtest=25 überprüft wird 2. Falls das Zeichen im Q-Register eine Null (0) ist, wird per CJP 000 zurück zum IFETCH gesprungen, da das Wort hier zuende ist.
4. Takt: 013	1. Nun wird die Konstante 0x0041 vom dem im Q-Register gspeicherten Zeichen abgezogen, jedoch nicht wieder ins Q-Register geschrieben, da das Zeichen für weitere Vergleiche unverändert bleiben muss. 2. Dabei werden die entsprechenden Flags des Microstatusregisters geändert (CEu=L)
5. Takt: 014	1. Durch die vorherige Subtraktion wurden die Flags für den Vergleich größer gleich 41 (entspricht dem Großbuchstaben A) gesetz und mit srtest 32 überprüft. 2. Sollte das Zeichen kleiner sein, wird mit CJP 01A gesprungen.
6. Takt: 015	1. Nun wird die Konstante 0x005A vom dem im Q-Register gspeicherten Zeichen abgezogen, jedoch nicht wieder ins Q-Register geschrieben, da das Zeichen für weitere Vergleiche unverändert bleiben muss. 2. Dabei werden die entsprechenden Flags des Microstatusregisters geändert (CEu=L)
7. Takt: 016	1. Durch die vorherige Subtraktion wurden die Flags für den Vergleich kleiner gleich 5A (entspricht dem Großbuchstaben Z) gesetz und mit srtest 34 überprüft. 2. Sollte das Zeichen größer sein, wird mit CJP 01A gesprungen.
8. Takt: 017	1. Da wir nun wissen, daß das im Q-Register gespeicherte zeichen ein Großbuchstabe sein muss, addieren wir 0x0020, was dem Umwandeln eines Großbuchstabens in einen Kleinbuchstaben entspricht. 2. Da wir das Zeichen jedoch noch in den Hauptspeicher zurückschreiben müssen, wird das Ergebnis im Q-Register zwischengespeichert.
9. Takt: 018	1. Nun wird die Adresse des Zeichens, welches im Register RB steht, auf den Adressbus gelegt und per MWE=W eine Schreibanweisung gegeben.
10. Takt: 019	1. Das veränderte Zeichen aus dem Q-Register wird nun auf dem Datenbus gelegt und so an seine Ursprüngliche Adresse im Hauptspeicher geschrieben.
11. Takt: 01A	1. Die Adresse im Register RB wird um 1 erhöht und es wird zu der Microprogrammadresse 020 gesprungen, um das nächste Zeichen zu verarbeiten.

(2) Microprogramm *strtolower_imm*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
strtolower_imm																											
020	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CONT	020	H	H	I	E	H	R
imm_in_r0_laden																											
021	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	RAMF	7	MR	0	MR	H	H	CI0	X	L	H	77	PS	CJP	012	H	H	H	H	H	R
strtolower_[R0]																											
022	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	NOP	7	MR	0	MR	AB	H	CI0	X	L	H	77	PS	CONT	010	H	H	H	H	H	R
023	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	SUBS	QREG	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	000	H	H	H	H	H	R
024	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	f	MR	f	MR	H	H	CI1	X	H	H	25	C	CJP	000	H	H	H	H	H	R
025	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	0041	DQ	SUBR	NOP	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	000	H	H	H	H	H	R
026	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	7	MR	f	MR	H	H	CI0	X	H	H	32	C	CJP	02C	H	H	H	H	H	R
027	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	005A	DQ	SUBR	NOP	7	MR	f	MR	H	H	CI1	X	L	H	77	PS	CONT	015	H	H	H	H	H	R
028	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	OR	NOP	7	MR	f	MR	H	H	CI0	X	H	H	34	C	CJP	02C	H	H	H	H	H	R
gross->klein																											
029	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	0020	DQ	ADD	QREG	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CONT	017	H	H	H	H	H	R
02a	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	NOP	7	MR	f	IR	AB	H	CI0	X	H	H	77	PS	CONT	018	H	H	H	H	H	W
02b	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZQ	ADD	NOP	7	MR	f	MR	H	DB	CI0	X	H	H	77	PS	CONT	000	H	H	H	H	H	R
R0+1																											
02c	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	RAMF	7	MR	0	MR	H	H	CI1	X	H	H	77	PS	CJP	022	H	H	H	H	H	R

strtolower_imm	Wandelt alle vorkommenden Kleinbuchstaben eines Wortes ab der Adresse des Immediates in Großbuchstaben um (Ende eines Wortes ist 0x0000 (NULL))
1. Takt: 020	1. Durch BZ_EA=E die Adresse des Immediate auf den Adressbus gelegt. 2. Der Befehlszähler wird um 1 erhöht
2. Takt: 021	1. Am K-Mux liegt nun der Wert des Immediates an, welcher nach R0 geladen wird
3. Takt: 022	1. R0 wird auf den Adressbus gelegt, um das erste Zeichen des Wortes zu erhalten
4. Takt: 023	1. Am K-Mux liegt nun der Wert des ersten Zeichens an, welches nach einer Subtraktion mit 0 in das Q-Register geschrieben wird 2. Dabei werden die entsprechenden Flags des Microstatusregisters geändert (CEu=L)
5. Takt: 024	1. Durch die Vorherige Subtraktion wurden bereits die Flags für einen Vergleich gleich 0 gesetzt, was nun mit srtest=25 überprüft wird. 2. Falls das Zeichen im Q-Register eine Null (0) ist, wird per CJP 000 zurück zum IFETCH gesprungen, da das Wort hier zuende ist.
6. Takt: 025	1. Nun wird die Konstante 0x0041 vom dem im Q-Register gspeicherten Zeichen abgezogen, jedoch nicht wieder ins Q-Register geschrieben, da das Zeichen für weitere Vergleiche unverändert bleiben muss. 2. Dabei werden die entsprechenden Flags des Microstatusregisters geändert (CEu=L)
7. Takt: 026	1. Durch die vorherige Subtraktion wurden die Flags für den Vergleich größer gleich 41 (entspricht dem Großbuchstaben A) gesetz und mit srtest 32 überprüft. 2. Sollte das Zeichen kleiner sein, wird mit CJP 02C gesprungen.
8. Takt: 027	1. Nun wird die Konstante 0x005A vom dem im Q-Register gspeicherten Zeichen abgezogen, jedoch nicht wieder ins Q-Register geschrieben, da das Zeichen für weitere Vergleiche unverändert bleiben muss. 2. Dabei werden die entsprechenden Flags des Microstatusregisters geändert (CEu=L)
9. Takt: 028	1. Durch die vorherige Subtraktion wurden die Flags für den Vergleich kleiner gleich 5A (entspricht dem Großbuchstaben Z) gesetz und mit srtest 34 überprüft. 2. Sollte das Zeichen größer sein, wird mit CJP 02C gesprungen.
10. Takt: 029	1. Da wir nun wissen, daß das im Q-Register gespeicherte zeichen ein Großbuchstabe sein muss, addieren wir 0x0020, was dem Umwandeln eines Großbuchstabens in einen Kleinbuchstaben entspricht. 2. Da wir das Zeichen jedoch noch in den Hauptspeicher zurückschreiben müssen, wird das Ergebnis im Q-Register zwischengespeichert.
11. Takt: 02A	1. Nun wird die Adresse des Zeichens, welches im Register R0 steht, auf den Adressbus gelegt und per MWE=W eine Schreibanweisung gegeben.
12. Takt: 02B	1. Das veränderte Zeichen aus dem Q-Register wird nun auf dem Datenbus gelegt und so an seine Ursprüngliche Adresse im Hauptspeicher geschrieben.
13. Takt: 02C	1. Die Adresse im Register R0 wird um 1 erhöht. 2. Es wird zu der Microprogrammadresse 022 gesprungen, um das nächste Zeichen zu verarbeiten.

(3) Microprogramm *add_imm_RB*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
add_imm_rb																											
030	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	QREG	7	MR	f	IR	H	H	CI0	X	H	H	77	PS	CONT	030	H	H	I	E	H	R
031	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DQ	ADD	RAMF	7	MR	f	IR	H	H	CI0	X	H	H	77	PS	JZ	031	H	H	H	H	H	R

add_imm_rb	auf das Register RB wird ein Immediate addiert
1. Takt: 030	1. Der Inhalt von RB wird ins Q Register geschrieben 2. Der Befehlszähler wird um 1 erhöht und der Inhalt auf den Adressbus ausgegeben (Immediate)
2. Takt: 031	1. Der Immediate liegt am Datenbus an und wird mit dem Inhalt des Q Registers addiert und nach RB zurückgeschrieben 2. Sprung zu IFETCH

(4) Microprogramm cmp_*imm_RA*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
cmp_imm_RA_(ist_RA_grösser/kleiner_als_imm)																											
080	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZA	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CONT	080	H	H	H	E	H	R
081	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DA	SUBR	QREG	7	IR	f	MR	H	H	CI1	X	H	L	77	PS	JZ	29E	H	H	I	H	H	R

cmp_imm_RA	vergleicht einen Immediate mit dem Inhalt von RA und setzt die entsprechenden Flags im Statusregister
1. Takt: 080	1. Befehlszähler auf Adressbus ausgeben
2. Takt: 081	1. Der Immediate liegt am Datenbus an 2. Mit SUBR wird dieser von RA abgezogen, CI=1 muss dabei gesetzt sein 3. Das Maschinenstatusregister wird entsprechend des Ergebnisses verändert (CEM=L) 4. Es erfolgt ein Sprung zu IFETCH durch JZ, der Befehlszähler wird noch um 1 erhöht

(5) Microprogramm *mov_RA_[RB]*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
mov_RA_[RB]																											
050	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZB	ADD	NOP	7	MR	f	IR	AB	H	CI0	X	H	H	77	PS	CONT	050	H	H	H	H	H	W
051	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZA	ADD	NOP	7	IR	f	MR	H	DB	CI0	X	H	H	77	PS	CJP	000	H	H	H	H	H	R

move_RA_[RB]	der Inhalt von RA wird an die Speicheradresse RB geschrieben
1. Takt: 050	1. RB wird auf den Adressbus ausgegeben 2. Im nächsten Takt soll in den Hauptspeicher geschrieben werden (MWE=W)
2. Takt: 051	1. Die zu schreibenden Daten in RA werden auf den Datenbus ausgegeben und somit an die Speicheradresse RB geschrieben 2. Der Schreibzugriff wird beendet (MWE=R) 3. Es erfolgt ein Sprung zu IFETCH

(6) Microprogramm *mov_[RA]_RB*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
move_[RA]_RB																											
040	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	ZA	ADD	QREG	7	IR	f	MR	AB	H	CI0	X	H	H	77	PS	CONT	040	H	H	H	H	H	R
041	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	RAMF	7	MR	f	IR	H	H	CI0	X	H	H	77	PS	CJP	000	H	H	H	H	H	R

move_[RA]_RB	der Inhalt der Speicheradresse in RA wird nach RB kopiert
1. Takt: 040	1. Der Inhalt von RA wird auf den Adressbus ausgegeben
2. Takt: 041	1. Der Inhalt der Speicheradresse RA liegt am Datenbus an und wird nach RB geschrieben 2. Es erfolgt ein Sprung zu IFETCH

(7) Microprogramm *jmp_imm*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
jmp_imm																											
020	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CONT	020	H	H	H	E	H	R
021	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	CJP	000	L	H	H	H	H	R

jmp_imm	springt an die als Immediate angegebene Adresse
1. Takt: 020	1. Der Immediate wird auf den Adressbus gelegt (BZ_EA=E)
2. Takt: 021	1. Das Sprungziel liegt am Datenbus an und wird durch BZ_LD=L in den Befehlszähler geladen 2. Es wird zu IFETCH gesprungen

(8) Microprogramm *jmpz_imm*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
jmpz_imm																											
060	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	45	C	CJP	062	H	H	I	E	H	R
061	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	061	H	H	H	H	H	R
062	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	062	L	H	H	H	H	R

jmpz_imm	springt an die als Immediate angegebene Adresse wenn bei der vorherigen Operation A = B
1. Takt: 060	1. Maschinenstatusregister wird abgefragt ob A = B, also ob im Statusregister Z=1 (SRTEST=45) 2. Ist dies der Fall erfolgt ein Sprung nach 062 3. Der Befehlszähler wird um 1 erhöht und somit durch BZ_EA=E der Immediate auf den Adressbus gelegt
2. Takt: 061	1. Es erfolgt ein Sprung zu IFETCH durch JZ
3. Takt: 062	1. Das Sprungziel liegt am Datenbus an und wird durch BZ_LD=L in den Befehlszähler geladen 2. Per JZ wird zu IFETCH gesprungen und der nächste Befehl von der angegebenen Adresse geladen

(9) Microprogramm *jpgt_imm*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
jpgt_imm																											
010	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	54	C	CJP	012	H	H	I	E	H	R
011	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	011	H	H	H	H	H	R
012	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	012	L	H	H	H	H	R

jpgt_imm	springt an die als Immediate angegebene Adresse wenn bei der vorherigen Operation A > B
1. Takt: 010	1. Maschinenstatusregister wird abgefragt ob A > B, also ob im Statusregister C=1 und Z=0 (SRTEST=54) 2. Ist dies der Fall erfolgt ein Sprung nach 012 3. Der Befehlszähler wird um 1 erhöht und somit durch BZ_EA=E der Immediate auf den Adressbus gelegt
2. Takt: 011	1. Es erfolgt ein Sprung zu IFETCH durch JZ
3. Takt: 012	1. Das Sprungziel liegt am Datenbus an und wird durch BZ_LD=L in den Befehlszähler geladen 2. Per JZ wird zu IFETCH gesprungen und der nächste Befehl von der angegebenen Adresse geladen

(10) Microprogramm *jplt_imm*

	Interrupt		Am2901											Am2904					Am2910			BZ				MEM	
jplt_imm																											
070	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	52	C	CJP	072	H	H	I	E	H	R
071	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	K	FFFF	AQ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	071	H	H	H	H	H	R
072	ie	int	kmux	cons	src	func	dest	raadr	asel	rbadr	bsel	abus	dbus	cinmux	shcrtl	CEu	CEM	srtest	ccen	ins	bar	ld	ed	inc	ea	irld	mwe
	DIS	ENI	D	FFFF	DZ	ADD	NOP	7	MR	f	MR	H	H	CI0	X	H	H	77	PS	JZ	072	L	H	H	H	H	R

jplt_imm	springt an die als Immediate angegebene Adresse wenn bei der vorherigen Operation A < B
1. Takt: 070	1. Maschinenstatusregister wird abgefragt ob A < B, also ob im Statusregister C=0 (SRTEST=52) 2. Ist dies der Fall erfolgt ein Sprung nach 072 3. Der Befehlszähler wird um 1 erhöht und somit durch BZ_EA=E der Immediate auf den Adressbus gelegt
2. Takt: 071	1. Es erfolgt ein Sprung zu IFETCH durch JZ
3. Takt: 072	1. Das Sprungziel liegt am Datenbus an und wird durch BZ_LD=L in den Befehlszähler geladen 2. Per JZ wird zu IFETCH gesprungen und der nächste Befehl von der angegebenen Adresse geladen