

AVSIM 51

Allgemeines

Kommandozeilenaufruf

(1) AVSIM51

Es meldet sich eine Auswahl über insgesamt 6 mögliche Prozessoren:

A:8051/8751, B:8052/8752, C:8031, D:8032, E:80C51, F:80C31

(2) AVSIM51 <argument>

wobei <argument> eine beliebige Folge von Tastenberührungen ist.

Beispiele:

AVSIM51 A A überspringt das erste Menu und wählt 8051.

AVSIM51 AFLDEMO.CMD A überspringt das erste Menu und wählt 8051. F Menu-Punkt commandFile. L Load Lädt die Kommando-Datei DEMO.CMD. Die Kommando-Datei enthält dann weitere Tastaturkommandos, die dann automatisch am Bildschirm ablaufen.

(3) SIM

Am einfachsten ruft man den Simulator mit der Batchdatei SIM.BAT auf. (Sie wählt eine 8031 CPU (ROM-less) als Simulationsobjekt.)

AVSIM-kompatible Dateien

Dateien im HEX-Format können durch den AVSIM bearbeitet werden. Sowohl der Assembler X8051 als auch der Assembler ASM51 sind dazu geeignet. Der Aufruf erfolgt wie folgt:

```
X8051/LINK ASM <name>
ASM51/OH ASS <name>
```

In beiden Fällen hat die Datei den Namen <name>.ASM. Soll auch eine Symboldatei <name>.SYM verwendet werden, muß auf das durch AVSIM vorgegebene SYM-Format geachtet werden.

Betriebsarten

Der Simulator unterscheidet zwischen **Kommando-Modus** (Menü) und **Display-Modus** (Screen). Mit <Esc> schaltet man zwischen Menü und Screen um. Um den Simulator zu verlassen, wählen Sie im Menü-Modus den Punkt Quit an. Danach geben Sie E für Exit ein. Die beiden Modi werden im Folgenden erläutert.

Kommando-Modus(Menü)

Mit <Space> schaltet man zwischen den beiden Menü-Balken um. Um zwischen den Menüpunkten zu wählen, benutzt man die Cursor-Tasten. Angewählt wird ein Menüpunkt entweder durch Cursor-Auswahl und <Enter> oder durch die entsprechende (großgeschriebene) Taste. (Auch die Menü-Punkte des versteckten Menü-Balkens.) Mit <Ctrl-C> kommt man jederzeit zum Hauptmenü. Mit <Esc> schaltet man jederzeit zwischen Menü und Screen um. Um Hilfe zu bekommen ruft man vom Hauptmenü aus den Menüpunkt Help auf. Es erscheint ein Untermenü, in dem man die 3 verschiedenen Helps aufrufen kann.

Display-Modus(Bildschirm)

Im linken Teil sieht man den disassemblierten Op-Code der nächsten auszuführenden Befehle. Im rechten Teil sind die Special-Function-Registers der zu simulierenden CPU abgebildet. Die beiden Dumpareas geben, umschaltbar (Pg-Up, Pg-Dn), einen Teil des internen RAMs wieder. Außerdem werden die Zustände der umschaltbaren Simulator - Funktionen angezeigt (siehe Funktionstasten).

Speicherbereiche

Das Festlegen des definierten Speicherbereichs erfolgt vor dem Aufrufen des Simulators und zwar durch die Datei SIM.SYM. Sie ist eine normale ASCII-DATEI, die wie folgt aufgebaut ist:

```
SG CODE C:0000 1000
SG DATA X:8000 A000
```

Die erste Zeile definiert das ROM (in obigem Beispiel von 0000 bis 1000 (Hexadezimal)). Die zweite Zeile definiert das RAM (in obigem Beispiel von 8000 bis A000 (Hexadezimal)). Startet man den Simulator mit der Batch-Datei, wird diese Datei automatisch geladen. Ansonsten muß man Sie im Load-Menü als SymbolTable laden. Definierter Speicherbereich wird bei einschalten mit 00h gefüllt, nicht definierter Speicherbereich mit FFh.

Hinweis: Zu großer definierter Speicher benötigt Platz auf der Diskette! ==> Zugriff dauert lange!

Ein Programm laden

Ein zu simulierendes Programm muß, um es in den Simulator zu laden im Intel-Hex-Format (Quelltext ==> Assembler(ASM51) ==> Objectcode_to_Hex-Conversion(OH) ==> Intel-Hex-File) sein. Vom Hauptmenü aus wählt man LOAD und dann PROGRAM.

Simulation der seriellen Schnittstelle

Die zu empfangenden Zeichen müssen in einer ASCII-Datei gespeichert sein. (Achtung: Auch Carrige-Return und Line-Feed gelten als Zeichen.)

Im Menü IO wählt man OPEN und gibt den Namen der Eingabe-Datei an. Dann den der Ausgabe-Datei. (Einer der Datei-Namen kann mit <ENTER> übersprungen werden.) Als Trigger verwendet man CYCLES. Anschließend gibt man an nach wievielen Zyklen ein Byte hereinkommt. Bei Format gibt man MAPBYTE an. Als 'Expression' gibt man SBUF an. (Dort wird empfangenes Zeichen abgespeichert.) Um den IO Transfer zu aktivieren muß man im SET Menü mit CYCLES den Cycle-Counter einschalten.

Hinweis zu Assembler ASM51

Am Ende des Quelltextes, der mit einem Texteditor erstellt wird, muß das Assembler-Directive 'END' stehen, das von einem Strichpunkt ';' gefolgt wird. Hinter diesem Strichpunkt darf K E I N Zeichen mehr stehen. (Auch nicht Carrige-Return, Line-Feed oder ein Blank!)

Bildschirmbild

```
8051/8751 AVSIM 8051 Simulator/Debugger V1.22
CPU REGISTERS      FLAGS      SCL SPD DSP SKP CURSOR
C Accumulator:    AC F0 OV P OFF HI ON OFF MENU
C 00000000:00:~  0 0 0 0 0
Cycles:
addr      data
PC:0000 >1 FF FF FF FF Timers TH/TL TF/TR G/T/M1/M0
SP: 07 > 00 00 00 00 T0: 00 00 0 0 0 0 0 0
        00 00 00 00 T1: 00 00 0 0 0 0 0 0
DP:0000 > FF FF FF FF
R0:00:~ > 00:~ RB:00 Ints A S Tl Xl T0 X0 Edg IT IE
R1:00:~ > 00:~ B:00 En 0 0 0 0 0 0 X0: 0 0
R2:00 R4:00 R6:00 Pr 0 0 0 0 0 0 X1: 0 0
R3:00 R5:00 R7:00 SBUF: In Out PCON:0xxxxxxx
Data Space 00:~ 00:~ SCON:00000000
0000 00 00 00 00 00 00 00 00 ----- Ports
0008 00 00 00 00 00 00 00 00 ----- P0 11111111
0010 00 00 00 00 00 00 00 00 ----- FF:~:11111111
0018 00 00 00 00 00 00 00 00 ----- P1 11111111
Data Space FF:~:11111111
0020 00 00 00 00 00 00 00 00 ----- P2 11111111
0028 00 00 00 00 00 00 00 00 ----- FF:~:11111111
0030 00 00 00 00 00 00 00 00 ----- P3 11111111
0038 00 00 00 00 00 00 00 00 ----- FF:~:11111111
Select Command- or use arrow-key
Dump Expression commandFile Help IOLoad --space--
```

Kommandozeile

Die folgende Übersicht zeigt den gesamten Menübaum des AVSIM51, die Einrückung gibt die entsprechende Ebene an. Rechts von der Kurzbezeichnung steht, durch Bindestriche ge-

trennt, der Kurzcode, mit dem man das Kommando erreicht. Dabei bedeuten [...] eines aus, <n> Dateiname, <c> condition, <a> Adresse, <d> Daten, <e> expression.

Dump -----D

Einstellung des Datenbereiches, des in den zwei Datenfenstern eingestellt werden kann. Durch die Wahl *Indirect* können auch Datenbereiche im externen RAM oder ROM dargestellt werden.

Data Space 1 -----D1

Absolute -----D1A<a>

Eingabe Startadresse

Indirect -----D1I<a>

Zeiger auf indizierte Speicherzelle

Data Space 2 -----D2

Absolute -----D2A<a>

Eingabe Startadresse

Indirect -----D2I<a>

Zeiger auf indizierte Speicherzelle

Expression -----E

Eingabe eines Ausdrucks in jene Bildschirmzelle, auf der im Augenblick der Cursor(PC) für steht (siehe ENTER EXPRESSION)

commandFile -----F

Man kann eine oft wiederkehrende Folge von Tastendrücker abspeichern (Makro) und dann wieder abrufen.

Load -----FL<n>

Wiedergabe einer gespeicherten Tastenfolge (Makro)

Open -----FO<n>

Eröffnen einer Makro-Datei für Tastendrucke. Jetzt führt man die zu sichernde Folge von Befehlen aus. Um das Makro zu sichern, wählt man wieder *commandFile* und dann *Close*. Hat man sich geirrt, kann man statt *Close* *Restart* wählen und ohne Eingabe eines neuen Namens von vorne beginnen.

Achtung: Es werden auch die Tastendrucke, die zum Speichern des Makros notwendig sind, aufgezeichnet. D.h. nachdem ein Makro aufgeführt wurde, befindet man sich im *commandFile*-Menü.

Close -----FC

Schließen der Datei für Tastendrucke

Restart -----FR

Neubeginn

Help -----H

Hilfebildschirme

Commands -----HC

Besonderheiten der Kommandozeile.

Display -----HD

Bildschirmsteuerung

Simulation -----HS

Funktionstastenbelegung

Avocet

Herstelleradresse, sowie Hinweise auf Aufrufparameter bezüglich Bildschirmkarte und Hinweise auf andere Produkte des Herstellers.

IO-File ----- I

Der Inhalt einer Datei kann den Inhalt einer Prozessoradresse beeinflussen. Ebenso kann der Inhalt einer Prozessoradresse in eine Datei gespeichert werden. Der Dateizugriff (Speichern oder Lesen) erfolgt entweder bei jedem Zugriff durch einen Befehl oder nach einer einstellbaren Anzahl von Prozessortaktzyklen.

Open----- IO<n>[YN]<n>[CO]<a>[IO]
Eingabedatei <n>, Wiederholen der Datei bei Dateieinde [YN], Ausgabedatei <n>, Triggerung auf Taktzyklen oder Befehlszugriff [CO], Anbindung an adresse <a>, Richtung In/Out [IO].

Close----- IC

Load ----- L

Laden von Dateien

Avocet ----- LA<n>

Data ----- LD<n>

Laden von RAM-Daten.

Program ----- LP<n>

name.HEX

Symbol-table ----- LS<n>

Laden einer Symbol-Tafel.

rOm ----- LO<n>

Memory ----- M

Bearbeitung des RAM-Speichers.

Clear ----- MC<a><a>

Startadresse, Stopadresse.

Fill ----- MF<a><a><d>

Startadresse, Stopadresse, Füllwert.

Move ----- MM<a><a><a>

Startadresse, Stopadresse, Zieladresse.

Search ----- MS<a><a><d>

Startadresse, Stopadresse, Suchbegriff.

searchNext ----- MN

Wiederholen der letzten Suche.

Patch ----- P

Direkte Eingabe von Mnemonics

Patch Code ----- PP

Open ----- PO<n>

Datei zum Mitschreiben händisch assemblierter Zeilen.

Close ----- PC

Mitschreiben abschließen

Quit ----- Q

Exit ----- QE

Reset ----- R

Dient zum Rücksetzen verschiedener Funktionen. Unter anderem kann ein Reset der CPU simuliert werden, sowie der Cycles-Counter zurückgesetzt werden.

Cpu ----- RC

Disptrace ----- RD

cYcles ----- RY

Set ----- S

Hier können Breakpoints gesetzt, der Cycles-Counter ein/ausgeschaltet, die Chip-Konfiguration (P2 für Higher-Adressen) und andere Funktionen eingestellt werden.

Achtung: In diesem Unter-Menü gibt es 2 Menü-Balken(mit SPACE umschaltbar!).

Memory Map ----- SM

random-Access ----- SMA

lower Address - higher Address

read-Only ----- SMO

lower Address - higher Address

Passpoint ----- SP

R/W-point ----- SPA

R/W-range ----- SPB

Write-point ----- SPC

Write-range ----- SPD

options ----- ST

Page ----- STP[YN]

Change external DataDrive with Port P2 from No to: [YN]

cYcles ----- SY

V-drive ----- SV

Set virtual memory

Set Breakpoint ----- S__

On Breakpoints, a delay can be specified by typing digits BEFORE selecting the Breakpoint TYPE. On sticky breakpoints, this delay is restored after each activation.

Conditional ----- S__C

Trap when condition match.

Indirect ----- S__CI<e><c>

Mask ----- S__CM[01x]

Range ----- S__CR<e><e><c>

Value ----- S__CV<e><c>

Dynamic ----- S__D

Cleared automatically upon Trap

R/W-point ----- S__DA

R/W-range ----- S__DB

Write-point ----- S__DC

Write-range ----- S__DD

Opcode ----- S__O<mnem>

Sticky ----- S__S

R/W-point ----- S__SA

R/W-range ----- S__SB

Write-point ----- S__SC

Write-range ----- S__SD

setUp ----- U

Der AVSIM51 hat die Fähigkeit, ausgeführte Befehle rückgängig machen zu können. In diesem Menüpunkt legt man die Gedächtnislänge fest.

Undo ----- UU

View -----V

Dient zum Anzeigen verschiedener Simulator-Zustände.

Bkpts -----VB

BKPT Addr EndAddr Access Life Delay/count
Condition

IO-files -----VI

Memory-map -----VM

MEMORY MAP Label Addr EndAddr Type
Code Space: 0000 to FFFF
Data Space: 0000 to 00FF
R0: D:0000 007F
External Data Space: 0000 to FFFF

Opc-traps -----VO

Opcode Traps: Operation Address Mode
ADD A,R0

Passpts -----VP

PASS Addr EndAddr Access Count Condition

Symbols -----VS

Alpha -----VSA

Symbol	Addr	Symbol	Addr	Symbol	Addr	Symbol	Addr
AC	B:00D6	PX0	B:00B8	R5	D:0005	SM1	B:009E
CY	B:00D7	PX1	B:00BA	R5"	D:0015	SM2	B:009D
EA	B:00AF	R0	D:0000	R5'	D:000D	T0	B:00B4
ES	B:00AC	R0"	D:0010	R5\	D:001D	T1	B:00B5
ET0	B:00A9	R0'	D:0008	R6	D:0006	T2	B:0090
ET1	B:00AB	R0\	D:0018	R6"	D:0016	T2EX	B:0091
EX0	B:00A8	R1	D:0001	R6'	D:000E	TB8	B:009B
EX1	B:00AA	R1"	D:0011	R6\	D:001E	TF0	B:008D
EXTI0	C:0003	R1'	D:0009	R7	D:0007	TF1	B:008F
EXTI1	C:0013	R1\	D:0019	R7"	D:0017	TI	B:0099
F0	B:00D5	R2	D:0002	R7'	D:000F	TIMER0	C:000B
IE0	B:0089	R2"	D:0012	R7\	D:001F	TIMER1	C:001B
IE1	B:008B	R2'	D:000A	RB8	B:009A	TIMER2	C:002B
INT0	B:00B2	R2\	D:001A	RD	B:00B7	TR0	B:008C
INT1	B:00B3	R3	D:0003	REN	B:009C	TR1	B:008E
IT0	B:0088	R3"	D:0013	RESET	C:0000	TXD	B:00B1
IT1	B:008A	R3'	D:000B	RI	B:0098	WR	B:00B6
OV	B:00D2	R3\	D:001B	RS0	B:00D3		
P	B:00D0	R4	D:0004	RS1	B:00D4		
PS	B:00BC	R4"	D:0014	RXD	B:00B0		
PT0	B:00B9	R4'	D:000C	SINT	C:0023		
PT1	B:00BB	R4\	D:001C	SM0	B:009F		

Registers -----VSR

Symbol	Symbol	P2
ACC	SBUF	P2L
B	SBUF _L	P3
CYCLES	SCON	P3L
DP	SP	PC
DPH	T2CON	PCON
DPL	TCON	PSW
IE	TH0	RCAP2H
IIP	TH1	RCAP2L
IP	TH2	
P0	TL0	
P0L	TL1	
P1	TL2	
P1L	TMOD	

Numbers -----VSN

Code -----VSC

Addr Symbol
C:0000 RESET

C:0003 EXTI0
C:000B TIMER0
C:0013 EXTI1
C:001B TIMER1
C:0023 SINT
C:002B TIMER2

Data -----VSD

Addr	Symbol				
D:0000	R0	D:000A R2'	D:0014 R4"	D:0016 R6"	
D:0001	R1	D:000B R3'	D:0015 R5"	D:0017 R7"	
D:0002	R2	D:000C R4'		D:0018 R0\	
D:0003	R3	D:000D R5'		D:0019 R1\	
D:0004	R4	D:000E R6'		D:001A R2\	
D:0005	R5	D:000F R7'		D:001B R3\	
D:0006	R6	D:0010 R0"		D:001C R4\	
D:0007	R7	D:0011 R1"		D:001D R5\	
D:0008	R0	D:0012 R2"'		D:001E R6\	
D:0009	R1	D:0013 R3"'		D:001F R7\	

ext -----VSX

Bit -----VSB

Addr	Symbol	Addr	Symbol
B:0088	IT0	B:00AC	ES
B:0089	IE0	B:00AF	EA
B:008A	IT1	B:00B0	RXD
B:008B	IE1	B:00B1	TXD
B:008C	TR0	B:00B2	INT0
B:008D	TF0	B:00B3	INT1
B:008E	TR1	B:00B4	T0
B:008F	TF1	B:00B5	T1
B:0090	T2	B:00B6	WR
B:0091	T2EX	B:00B7	RD
B:0098	RI	B:00B8	PX0
B:0099	TI	B:00B9	PT0
B:009A	RB8	B:00BA	PX1
B:009B	TB8	B:00BB	PT1
B:009C	REN	B:00BC	PS
B:009D	SM2	B:00D0	P
B:009E	SM1	B:00D2	OV
B:009F	SM0	B:00D3	RS0
B:00A8	EX0	B:00D4	RS1
B:00A9	ET0	B:00D5	F0
B:00AA	EX1	B:00D6	AC
B:00AB	ET1	B:00D7	CY

SFR -----VSS

Addr	Symbol	Addr	Symbol
R:0080	P0	R:00CD	TH2
R:0081	SP	R:00D0	PSW
R:0082	DP	R:00E0	ACC
R:0083	DPH	R:00F0	B
R:0087	PCON		
R:0088	TCON		
R:0089	TMOD		
R:008A	TL0		
R:008B	TL1		
R:008C	TH0		
R:008D	TH1		
R:0090	P1		
R:0098	SCON		
R:0099	SBUF		
R:00A0	P2		
R:00A8	IE		
R:00B0	P3		
R:00B8	IP		
R:00C8	T2CON		
R:00CA	RCAP2L		
R:00CB	RCAP2H		
R:00CC	TL2		

eXecute -----X

Ausführen eines einzelnen Simulator-Befehls.

Folgende Opcode-Syntax ist einzuhalten:

{label:} operation {operands} {; comments}

Dateien für den Simulator

Die mit '*' gekennzeichneten Dateien sind nachfolgend kurz beschrieben.

```
ASM      BAT      39 09-19-90 11:27a *
ASS      BAT      728 01-30-89 12:01a *
AVSIM    TXT      10693 04-08-91 1:16p <- hier eingearbeitet
AVSIM    DFV
AVSIM51  EXE      94720 01-30-89 12:01a
AVSIM51  HLP      5938 01-30-89 12:01a <- hier eingearbeitet
AVSIM51  OVR      29370 01-30-89 12:01a
AVSIM51  REG      1192 04-16-91 8:41a <- hier eingearbeitet
DEMO     CMD      118 08-29-86 7:57p *
DEMO     MSG      313 08-29-86 7:57p *
DEMO1    ASM      4334 08-29-86 7:57p *
DEMO1    HEX      372 08-29-86 7:57p
DEMO1    SYM      333 08-29-86 7:57p
DEMO2    ASM      4639 10-16-90 10:12p *
DEMO2    HEX      424 10-16-90 10:12p
DEMO2    LST      11188 10-16-90 10:12p
DEMO2    OBJ      981 10-16-90 10:12p
DEMO2    SYM      1 10-16-90 10:12p
IOTRAN   DOC      3717 01-30-89 12:01a *
IOTRAN   EXE      16896 01-30-89 12:01a *
KILLEOF  COM      11659 12-20-87 11:28p *
PAR      DAT      5 10-24-90 12:25a *
README   343 10-17-90 12:33p
RUNME    BAT      852 01-30-89 12:01a *
RUN_DEMO BAT      239 08-29-86 7:57p *
SIM      BAT      32 01-30-89 12:01a *
SIM      CMD      21 01-30-89 12:01a *
SIM      SYM      45 01-30-89 12:01a *
```

ASM.BAT

Assemblieren und Linken einer Datei mit dem Assembler/Linker von 2500AD. Aufruf: `ASM <name>. <name>` wird ohne Extension angegeben. Die Datei `<name>` hat die Extension `.ASM` und heißt daher `<name>.ASM`.

ASS.BAT

Assemblieren und Linken einer Datei mit `ASM51` und `OH`.

DEMO.CMD

Automatischer Ablauf des Simulators bei Aufruf mit `AVSIM51 AFLDEMO.CMD`. Es ist die Abfolge der Tastendrucke, die nötig sind, um das Programm `DEMO.ASM` auszuführen. Da es sich um Tastendrucke und nicht um ASCII-Zeichen handelt, ist die Datei nicht darstellbar.

DEMO.MSG

`DEMO.MSG` enthält einen ASCII-Text. Dieser Text wird durch das Programm `DEMO.ASM` über die serielle Schnittstelle eingespielt.

DEMO1.ASM

Originalversion des AVSIM-DEMO-Programms. (Übertragung einer Datei über die serielle Schnittstelle.)

DEMO2.ASM

Wie `DEMO1.ASM` aber modifiziert für den X8051-Assembler.

IOTRAN.DOC, IOTRAN.EXE

Programm zur Anpassung Datentransformation an verschiedene Formate.

```
hb - hex-to-byte      file.AH --> file.AB
hw - hex-to-word     file.AH --> file.AW
hb2 - hex-to-2 byte  file.AH --> file.AB1 & file.AB2
hw2 - hex-to-2 word  file.AH --> file.AW1 & file.AW2
bh - byte-to-hex     file.AB --> file.AH
wh - word-to-hex     file.AW --> file.AH
b2h - 2 byte-to-hex file.AB1 & file.AB2 --> file.AH
w2h - 2 word-to-hex  file.AW1 & file.AW2 --> file.AH
wd - word-to-decimal file.AW --> file.AD
b2d - 2 byte-to-decimal file.AB1 & file.AB2 --> file.AD
w2d - 2 word-to-decimal file.AW1 & file.AW2 --> file.AD
```

KILLEOF.COM

Löscht Dateieindemarkierung in einer Datei

PAR.DAT

Beispieldaten für die Anbindung an einen IO-Port.

RUNME.BAT

RUN_DEMO.BAT

Aufruf des Demo-Programms.

SIM.BAT

Aufruf von `AVSIM51`.

SIM.CMD

Abgespeicherte Folge von Tastendrucke.

SIM.SYM

Festlegung von Speicherbereichen.

Jedes Programm, in das sich ein Fehler einschleichen kann, wird auch einen enthalten. Folgerung: Jeder Fehler wird dort sitzen, wo er am spätesten entdeckt wird und den größtmöglichen Schaden anrichtet.