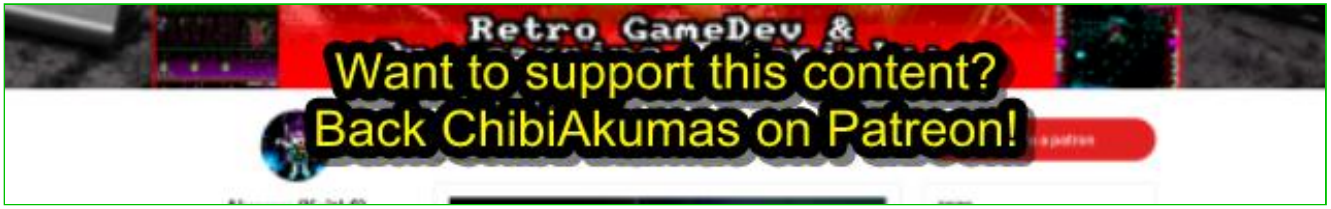


View Options
Default Dark
Simple (Hide this menu)
Print Mode (white background)

Top Menu
Main Menu
Youtube channel
Forum
AkuSprite Editor
Dec/Bin/Hex/Oct/Ascii Table

Z80 Content
Z80 Tutorial List
Learn Z80 Assembly
Hello World
Advanced Series
Multiplatform Series
Platform Specific Series
ChibiAkumas Series
Grime Z80
Z80 Downloads
Z80 Cheatsheet
Sources.7z
DevTools kit

Z80 Assembly programming for the Amstrad CPC

The CPC was the 8 bit I grew up with... while slower in some ways than the C64, it had far superior graphical capabilities to the ZX Spectrum, and usually beats the MSX for graphical speed because of it's smaller screen footprint (16k on the CPC to 24k on the MSX) and its CRTC graphics chip is favoured by the modern clever demo authors...

While the budget tape-based 464 machine had 64k, the disk system - the 6128 - has 128k and a disk system... and after market upgrades can give the machine up to 576k

There were two major generations of the CPC, the regular version, and the version released in the 90... the CPC+, which has has hardware sprites, enhanced color

	CPC	CPC+
Cpu	3.5mhz Z80	3.5mhz Z80

Ram	64k+	64k+
Vram	16k	128k
Resolution 4-color	320x200	256x212
Resolution 16-color	160x200	
Hardware Sprites	none	16 x 16x16 @ 16 color
Sound chip	AY	AY
Cartridge Rom	none	512k
Joystick	UDLR + 3 fire	UDLR + 2 fire

Z80 Platforms
Amstrad CPC
Elan Enterprise
Gameboy & Gameboy Color
Master System & GameGear
MSX & MSX2
Sam Coupe
TI-83
ZX Spectrum
Spectrum NEXT
Computers Lynx

ChibiAkumas Tutorials:

	Lesson H1 - Hello World on the CPC
	Lesson S1 - Easy Sprites on the CPC
	Lesson S11 - Joystick Reading on the Amstrad CPC
	Lesson P1 - Basic Firmware Text functions
	Lesson P2 - More Text Functions, Improvements... and the Sam Coupe!
	Lesson P3 - Bitmap graphics on the Amstrad CPC and Enterprise 128
	Lesson P6 - Keyreading on the Amstrad CPC, ZX Spectrum and Sam Coupe
	Lesson P13 - Palette definitions on the Amstrad CPC and CPC+
	Lesson P18 - Making Sound with the AY on the Amstrad CPC, MSX,ZX Spectrum.... and NeoGeo + Atari ST!!
	Lesson P24 - Bankswitching and hardware detection on the Amstrad CPC
	Lesson P32 - Hardware Sprites on the CPC+
	Lesson P35 - Playing Digital Sound with WAV on the AY!
	Lesson P36 - Playing Digital Sound with WAV on the CPC+ via DMA!
	Lesson P39 - Setting the CPC screen with CRTC registers
	Lesson P40 - Syncronized mode switches for 320x200 @ 16 color EGX graphics on the Amstrad CPC
	Lesson P41 - CRTC Rupture for Interrupt based splitscreen on the CPC
	Lesson P42 - Advanced CRTC Rupture

6502 Content
6502 Tutorial List
Learn 6502 Assembly
Advanced Series
Platform Specific Series
Hello World Series
Grime 6502
6502 Downloads
6502 Cheatsheet
Sources.7z
DevTools kit
6502 Platforms
Apple IIe
Atari 800 and 5200
Atari Lynx
BBC Micro
Commodore 64
Commander x16
Super Nintendo (SNES)
Nintendo NES / Famicom
PC Engine (Turbografx-16)
Vic 20

68000 Content

AY Sound Chip:

Register	Meaning	Bit Meaning	Details
0	Tone Pitch L - Channel A	LLLLLLLL	Lower value = Higher pitch
1	Tone Pitch H - Channel A	----HHHH	Lower value = Higher pitch
2	Tone Pitch L - Channel B	LLLLLLLL	Lower value = Higher pitch
3	Tone Pitch H - Channel B	----HHHH	Lower value = Higher pitch
4	Tone Pitch L - Channel C	LLLLLLLL	Lower value = Higher pitch
5	Tone Pitch H - Channel C	----HHHH	Lower value = Higher pitch
6	Noise Generator	---NNNNN	Higer = Faster noise
7	Mixer	--NNNTTT	N=Noise T=Tone (Channel --CBACBA 1=mute 0=normal)
8	Amplitude - Channel A	---EVVVV	E=Envelope (1=Enabled) VVVV=Volume
9	Amplitude - Channel B	---EVVVV	E=Envelope (1=Enabled) VVVV=Volume
10	Amplitude - Channel C	---EVVVV	E=Envelope (1=Enabled) VVVV=Volume
11	Envelope L (Volume over time)	LLLLLLLL	Lower=Faster Envelope
12	Envelope H (Volume over time)	HHHHHHHH	Lower=Faster Envelope
13	Envelope Selection	----EEEE	Envelope number (See PDF)

For more details, please see the **AY sound chip PDF**

Gate Array:

The Gate array is at port &7Fxx... It has multiple purposes depending on the top two bits passed in the C byte

7	6	5	4	3	2	1	0	Name	Bit meanings
0	0	-	B	P	P	P	P	Pen Selection	B=Border P=Pen (0-3 / 0-15)
0	1	-	C	C	C	C	C	Palette color selection	C= Color number (0-26)
1	0	-	I	H	L	M	M	Rom / Mode	I= Interrupt mode H=High rom bank L=Low rom bank M=screen mode

68000 Tutorial List
Learn 68000 Assembly
Hello World Series
Platform Specific Series
Grime 68000
68000 Downloads
68000 Cheatsheet
Sources.7z
DevTools kit
68000 Platforms
Amiga 500
Atari ST
Neo Geo
Sega Genesis / Mega Drive
Sinclair QL
X68000 (Sharp x68k)

8086 Content
Learn 8086 Assembly
Platform Specific Series
Hello World Series
8086 Downloads
8086 Cheatsheet
Sources.7z
DevTools kit
8086 Platforms
Wonderswan
MsDos

ARM Content
Learn ARM Assembly
Platform Specific Series
ARM Downloads
ARM Cheatsheet
Sources.7z
DevTools kit

1	1	B	B	B	R	R	R	Ram Bank	R = Ram config B= Bank number (0=128k 1+2=256k etc)

CRTC

The CRTC registers are selected with &BCxx (where xx is the register number)... and set with &BDxx (where xx is the new value

Hight Byte Value	F	E	D	C	B	A	9	8	Name
&BC	1	0	1	1	1	1	0	0	Register Select
&BD	1	0	1	1	1	1	0	1	Register Write

CRTC - Registers

CRTC registers will configure the Logical screen we draw on, and the physical screen shown by the Analog monitor...
You'll want to leave many of the settings alone, as they'll just result in a corrupt screen you can't view!

Reg Num	Name	Range	Bits	Default Value	Speccy 256x192	Overscaln 384x272(26k)	Details
&00	Horizontal Total	0-255	DDDDDDDD	63	63	63	Physical width of screen Leave alone!
&01	Horizontal Displayed	0-255	DDDDDDDD	40	32	48	Logical width in Chars (8 pixels in mode 1)
&02	Horizontal Sync Position	0-255	DDDDDDDD	46	42	51	Logical Xpos
&03	Horizontal and Vertical Sync Widths	0-15,0-15	VVVVHHHH	142	134	142	Physical width of screen Leave alone!
&04	Vertical Total	0-127	-DDDDDDD	38	38	38	Physical height of screen Leave alone!
&05	Vertical Total Adjust	0-31	---DDDDD	0	0	0	Scanline Offset
&06	Vertical Displayed	0-127	-DDDDDDD	25	24	34	Logical Height in Chars (8 Pixels)
&07	Vertical Sync position	0-127	-DDDDDDD	30	31	35	Logical Ypos of

							screen
&08	**Interlace and Skew**	0-3	------DD	0	0	0	0/2=off 1/3=on ? Leave alone!
&09	**Maximum Raster Address**	0-31	---DDDDD	7	7	7	Max Raster Address ? Leave alone!
&0A	**Cursor Start Raster**	0-127	-DDDDDDD	0	0	0	
&0B	**Cursor End Raster**	0-31	---DDDDD	0	0	0	
&0C	**Display Start Address (H)**	0-63	xxPPSSOO	00 / 16 / 32 / 48	00 / 16 / 32 / 48	12+1 / 28 / 44+1 / 60	PP=Screen Page (11=C000) S=Size(11=32k else 16k) O=Offset
&0D	**Display Start Address (L)**	0-255	OOOOOOOO	0	0	0	O=Offset
&0E	**Cursor Address (H)**	0-63	c--DDDDDD	0	0	0	
&0F	**Cursor Address (L)**	0-255	DDDDDDDD	0	0	0	
&10	**Light Pen Address (H)**	0-63	--DDDDDD	0	0	0	Read Only
&11	**Light Pen Address (L)**	0-255	DDDDDDDD	0	0	0	Read Only

Dragon 32/Tandy Coco
Fujitsu FM7
TRS-80 Coco 3
Vectrex

My Game projects
Chibi Aliens
Chibi Akumas

Work in Progress
Learn 65816 Assembly
Learn eZ80 Assembly

Misc bits
Ruby programming

CPC Plus hardware sprites!

The Amstrad CPC has 16 hardware sprites... each is 16x16

CPC+ sprites have a separate 16 colors to the normal palette, also they are always 16 color, even in mode 1!
CPC+ Sprite colors (-GRB) are defined by the range &6422-&643F

Sprite Number	DataAddr (256bytes)	Xpos (2bytes)	Ypos (2bytes)	Resolution (1byte)
1	&4000	&6000	&6002	&6004
2	&4100	&6008	&600A	&600C
3	&4200	&6010	&6012	&6014
4	&4300	&6018	&601A	&601C
5	&4400	&6020	&6022	&6024
6	&4500	&6028	&602A	&602C
7	&4600	&6030	&6032	&6034
8	&4700	&6038	&603A	&603C
9	&4800	&6040	&6042	&6044

10	&4900	&6048	&604A	&604C
11	&4A00	&6050	&6052	&6054
12	&4B00	&6058	&605A	&605C
13	&4C00	&6060	&6062	&6064
14	&4D00	&6068	&606A	&606C
15	&4E00	&6070	&6072	&6074
16	&4F00	&6078	&607A	&607C

DataAddr is the pointer to the sprite data... note only 4 bits of each byte are used (----CCCC).... so 16x16=256 bytes

Xpos is 2 bytes, in little endian format... screen co-ordinates are based on mode 2 - so xpos should be between -64 to +639 (0 is leftmost visible pixel)
Ypos is 2 bytes, in little endian format... Ypos should be between -64 to +200 (0 is topmost visible pixel)

Resolution is defined by a single byte:
PP
Each (X,Y) size can be 0-3... 0 will mean the sprite isn't shown, 1,2,3 are magnifications 1,2 and 4.... however please note, these are relative to Mode 2... so 'square' pixels are defined by %00001001 = 9
This is why You'll see in ChibiAkumas, sprites are set to 'on' with a value of 9, or 'off' with a value of 0

Keyboard Matrix

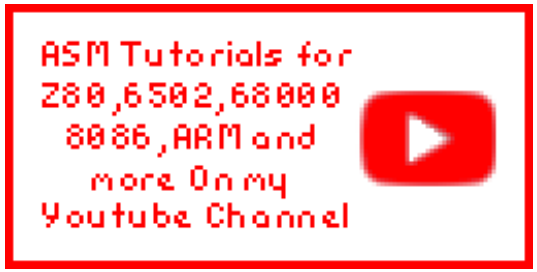
	0	1	2	3	4	5	6	7
&40	C-U	C-R	C-D	F9	F6	F3	F-ENT	F.
&41	C-L	CPY	F7	F8	F5	F1	F2	F0
&42	c	[	RET	]	4	SHIFT	\	c
&43	^	-	@	P	;	:	/	.
&44	0	9	O	I	L	K	M	,
&45	8	7	U	Y	H	J	N	
&46	6 J2-U	5 J2-D	R J2-L	T J2-R	G J2-F1	F J2-F2	B J2-F3	V
&47	4	3	E	W	S	D	C	X
&48	1	2	ESC	Q	TAB	A	CAPS	Z
&49	J1-U	J1-D	J1-L	J1-R	J1-F1	J1-F2	J1-F3	DEL

Amstrad Links

Winape - Not just the easiest to use CPC emulator, but the easiest Z80 platform for beginner ASM programmers!
www.cantrell.org.uk - A great source of CPC and ASM info.. My cheatsheet ASM list is based on the one from cantrell.org.uk
CRTC - Details of the amstrad CPC CRTC hardware
Amstrad Firmware guide - Pdf documenting the CPC firmware calls
CPC Firmware Guide - Detailed info on how the CPC hardware and firmware
Basic Manual - You'll want to know at least enough basic to do calls and operate the computer
CpcWiki - Web community full of helpful people!

General Z80 Assembly Tutorials:

B. Beginner series - Learn the basics
A. Advanced series - In more detail
M. Multiplatform series - programming methods that work on all systems



Recent New Content

[6809 Lesson 5 - More Maths - Logical Ops, Bit shifts and more]

[x68000 Hardware Sprites]

[Joypad & Pen on the GBA / NDS ... Key reading on Risc OS]

[C64 Hardware Sprites - 6502 ASM Lesson YQuest14]

[SNES Hardware sprites - 6502 ASM YQuest13]

[Vector drawing on the Vectrex]

[Graphics on the Fujitsu FM7]

CPC ASM: Tape loading on the Amstrad CPC (5K subs special)

68000 YQuest7 - Atari ST Specific code

Hardware Sprites on the NES - Lesson YQuest12

Hardware Sprites on the PC Engine / Turbografix

Joystick reading on the Vectrex - 6809 ASM

Gaming + more:

Emily The Strange (DS) - Live full playthrough

$150 calculator: Unboxing the Ti-84 Plus CE (eZ80 cpu)

Recent New Content

[6809 Lesson 5 - More Maths - Logical Ops, Bit shifts and more]

[x68000 Hardware Sprites]

Joypad & Pen on the GBA / NDS ... Key reading on Risc OS

C64 Hardware Sprites - 6502 ASM Lesson YQuest14

SNES Hardware sprites - 6502 ASM YQuest13

Vector drawing on the Vectrex

Graphics on the Fujitsu FM7

CPC ASM: Tape loading on the Amstrad CPC (5K subs special)

68000 YQuest7 - Atari ST Specific code

Hardware Sprites on the NES - Lesson YQuest12

Hardware Sprites on the PC Engine / Turbografix

Joystick reading on the Vectrex - 6809 ASM

Gaming + more:

Emily The Strange (DS) - Live full playthrough

$150 calculator: Unboxing the Ti-84 Plus CE (eZ80 cpu)

ASM Tutorials for
Z80,6502,68000
8086,ARM and
more On my
Youtube Channel

Questions,
Suggestions
Advice?
Discuss on the
Forums!

Recent New Content

[6809 Lesson 5 - More Maths - Logical Ops, Bit shifts and more]

[x68000 Hardware Sprites]

[Joypad & Pen on the GBA / NDS ... Key reading on Risc OS]

[C64 Hardware Sprites - 6502 ASM Lesson YQuest14]

[SNES Hardware sprites - 6502 ASM YQuest13]

[Vector drawing on the Vectrex]

[Graphics on the Fujitsu FM7]

[CPC ASM: Tape loading on the Amstrad CPC (5K subs special)]

[68000 YQuest7 - Atari ST Specific code]

Hardware Sprites on the NES - Lesson YQuest12

Hardware Sprites on the PC Engine / Turbografix

Joystick reading on the Vectrex - 6809 ASM

Gaming + more:

Emily The Strange (DS) - Live full playthrough

$150 calculator: Unboxing the Ti-84 Plus CE (eZ80 cpu)