Delve into the depths of Epson's HX-20 with a machine code disassembler — Elizabeth Wald explains.

The HX unravelled

The Epson HX-20 has a powerful monitor which enables the user easily to enter machine code programs. However, listing such programs can be laborious and slow, and requires you to be familiar with the hexadecimal codes. It is therefore useful to have a disassembler program to allow listings to be made using the standard 6301 machine code mnemonics.

6301 machine code is based on 6800 machine code with the addition of several new instructions (see table). Most of these additional instructions are the direct result of the ability to combine registers A and B to form a 16 bit register, known as D. Within register D the upper 8-bits are formed by A, and B forms the lower 8-bits. Any alteration to the contents of A or B will affect register D, and vice versa.

The instructions using register D are as follows: 'LDD' and 'STD' will load and store register D respectively. Addition and subtraction are done by 'ADD D' and 'SUB D' placing the result in D. All four instructions can be used on the immediate, direct, indexed or extended modes, with the obvious exception of 'STD' which cannot be used in the immediate mode. It is possible to perform an arithmetic shift left ('ASL D') and a logical shift right ('LSR D'). The contents of D can be exchanged with the contents of X using the 'XGDX' instruction.

Further instructions provided are a multiply instruction (MUL), which multiplies register A by register B, and places the result in register D, instructions to push or pull register X ('PSH X' and 'PUL X') and a 'sleep' instruction ('SLP'), which sends the processor into a low power 'sleep' mode.

The final group of additional instructions on the 6301 are the so-called

6301 INSTRUCTIONS NOT FOUND IN 6800

	IMPLIED	IMMEDIATE	DIRECT	INDEXED	EXTENDED
ADD D	—	C3	D3	E3	F3
ASL D	05	—	—	—	—
LDD	—	CC	DC	EC	FC
LSR D	04	—	—	—	—
MUL	3D	—	—	—	—
PSH X	3C	—	—	—	—
PUL X	38	—	—	—	—
SLP	1A	—	—	—	—
STD	—	—	DD	ED	FD
SUB D	—	83	93	A3	B3
XGDX	18	—	—	—	—

```
0B00 3C PSH X
0B01 36 PSH A
0B02 37 PSH B
0B03 07 TPA
0B04 36 PSH A
0B05 BF STS    0BD0
0B08 FE LDX    0BD4
0B0B 27 BEQ    0B18
0B0D BE LDS    0BD2
0B10 CC LDA D £0000
0B13 FD STA D  0BD4
0B16 6E JMP   X00
0B18 BE LDS    0BD2
0B1B 3B RTI
0B1C 01 NOP
0B1D 01 NOP
0B1E 01 NOP
0B1F 01 NOP
0B20 3C PSH X
0B21 36 PSH A
0B22 37 PSH B
0B23 07 TPA
0B24 36 PSH A
0B25 BF STS    0BD2
0B28 BE LDS    0BD0
0B2B 3B RTI
0B2C 01 NOP
0B2D 01 NOP
0B2E 01 NOP
0B2F 02 ***
0B30 86 LDA A £01
0B32 B7 STA A  0BFE
0B35 BD JSR    0B20
0B38 B6 LDA A  0BFF
0B3B 39 RTS
0B3C 01 NOP
0B3D 01 NOP
0B3E 8D BSR    0B30
0B40 37 PSH B
0B41 36 PSH A
0B42 B7 STA A  0BFF
0B45 4F CLR A
0B46 F6 LDA B  0B2F
0B49 F7 STA B  0BFE
0B4C 32 PUL A
0B4D 33 PUL B
0B4E 20 BRA    0B20
0B50 01 NOP
0B51 01 NOP
0B52 01 NOP
0B53 01 NOP
0B54 86 LDA A £FF
0B56 B7 STA A  0BFE
0B59 BE LDS    0BD0
0B5C 3B RTI
0B5D 01 NOP
0B5E 01 NOP
0B5F 01 NOP
0B60 8D BSR    0B3E
0B62 80 SUB A £30
0B64 81 CMP A £0A
0B66 25 BCS    0B6A
0B68 80 SUB A £07
0B6A 39 RTS
0B6B 8D BSR    0B60
0B6D 48 ASL A
0B6E 48 ASL A
0B6F 48 ASL A
0B70 48 ASL A
0B71 37 PSH B
0B72 36 PSH A
0B73 8D BSR    0B60
0B75 33 PUL B
0B76 1B ABA
0B77 33 PUL B
0B78 39 RTS
0B79 00 ***
0B7A 00 ***
0B7B 00 ***
0B7C 00 ***
0B7D 00 ***
0B7E 00 ***
0B7F 00 ***
0B80 37 PSH B
0B81 36 PSH A
0B82 8D BSR    0B6B
0B84 16 TAB
0B85 8D BSR    0B6B
0B87 36 PSH A
0B88 37 PSH B
0B89 38 PUL X
0B8A 32 PUL A
0B8B 33 PUL B
0B8C 39 RTS
0B8D 01 NOP
0B8E 01 NOP
0B8F 01 NOP
0B90 8B ADD A £30
0B92 81 CMP A £3A
0B94 25 BCS    0B98
0B96 8B ADD A £07
0B98 20 BRA    0B40
0B9A 36 PSH A
0B9B 36 PSH A
0B9C 44 LSR A
0B9D 44 LSR A
0B9E 44 LSR A
0B9F 44 LSR A
0BA0 8D BSR    0B90
0BA2 32 PUL A
0BA3 84 AND A £0F
0BA5 8D BSR    0B90
0BA7 32 PUL A
0BA8 39 RTS
0BA9 36 PSH A
0BAA 3C PSH X
0BAB 32 PUL A
0BAC 8D BSR    0B9A
0BAE 32 PUL A
0BAF 8D BSR    0B9A
0BB1 32 PUL A
0BB2 39 RTS
0BB3 00 ***
0BB4 00 ***
0BB5 00 ***
0BB6 00 ***
0BB7 00 ***
0BB8 36 PSH A
0BB9 86 LDA A £20
0BBB 8D BSR    0B40
0BBD 32 PUL A
0BBE 39 RTS
0BBF 01 NOP
0BC0 7F CLR    0B2F
0BC3 39 RTS
0BC4 36 PSH A
0BC5 86 LDA A £02
0BC7 B7 STA A  0B2F
0BCA 32 PUL A
0BCB 39 RTS
0BCC 01 NOP
0BCD 01 NOP
0BCE 01 NOP
0BCF 01 NOP
0F00 8E LDS   £0BEF
0F03 8D BSR    0F05
0F05 38 PUL X
0F06 18 XDX
0F07 83 SUB D £0305
0F0A FD STA D  0BF0
0F0D C3 ADD D £0100
0F10 FD STA D  0BF2
0F13 86 LDA A £3F
0F15 BD JSR    0BC4
0F18 01 NOP
0F19 01 NOP
0F1A 01 NOP
0F1B 01 NOP
0F1C 01 NOP
0F1D 01 NOP
0F1E 01 NOP
0F1F 01 NOP
0F20 01 NOP
0F21 01 NOP
0F22 01 NOP
0F23 01 NOP
0F24 01 NOP
0F25 01 NOP
0F26 01 NOP
0F27 86 LDA A £0D
0F29 BD JSR    0B40
0F2C 86 LDA A £0A
0F2E BD JSR    0B40
0F31 FE LDX    0BF4
0F34 BD JSR    0BA9
0F37 BD JSR    0BB8
0F3A A6 LDA A X00
0F3C 08 INX
0F3D FF STX    0BF4
0F40 BD JSR    0B9A
0F43 BD JSR    0BB8
0F46 C6 LDA B £02
0F48 3D MUL
0F49 F3 ADD D  0BF2
0F4C 18 XDX
0F4D FF STX    0BF8
0F50 A6 LDA A X00
0F52 C6 LDA B £03
0F54 3D MUL
0F55 F3 ADD D  0BF0
0F58 18 XDX
0F59 C6 LDA B £03
0F5B A6 LDA A X00
0F5D 08 INX
0F5E BD JSR    0B40
0F61 5A DEC B
0F62 26 BNE    0F5B
0F64 BD JSR    0BB8
0F67 FE LDX    0BF8
0F6A A6 LDA A X01
0F6C 36 PSH A
0F6D C6 LDA B £20
0F6F 84 AND A £70
0F71 27 BEQ    0F86
0F73 C6 LDA B £41
0F75 81 CMP A £20
0F77 27 BEQ    0F86
0F79 5C INC B
0F7A 81 CMP A £30
0F7C 27 BEQ    0F86
0F7E C6 LDA B £44
0F80 81 CMP A £40
0F82 27 BEQ    0F86
0F84 C6 LDA B £58
0F86 17 TBA
0F87 BD JSR    0B40
0F8A BD JSR    0BB8
0F8D 32 PUL A
0F8E 36 PSH A
0F8F 84 AND A £0C
0F91 C6 LDA B £20
0F93 4D TST A
0F94 27 BEQ    0FA3
0F96 C6 LDA B £23
0F98 81 CMP A £04
0F9A 27 BEQ    0FA3
0F9C 5C INC B
0F9D 81 CMP A £08
0F9F 27 BEQ    0FA3
0FA1 C6 LDA B £58
0FA3 17 TBA
0FA4 BD JSR    0B40
0FA7 FE LDX    0BF4
0FAA 32 PUL A
0FAB 81 CMP A £80
0FAD 24 BCC    0FDB
0FAF 84 AND A £03
0FB1 16 TAB
0FB2 5A DEC B
0FB3 27 BEQ    0FBD
0FB5 A6 LDA A X00
0FB7 BD JSR    0B9A
0FBA 08 INX
0FBB 20 BRA    0FB2
0FBD FF STX    0BF4
0FC0 BC CPX    0BF6
0FC3 22 BHI    0FCE
0FC5 FC LDA D  0BF0
0FC8 C3 ADD D £0322
0FCB 18 XDX
0FCC 6E JMP   X05
0FCE 86 LDA A £0D
0FD0 BD JSR    0B40
0FD3 86 LDA A £0A
0FD5 BD JSR    0B40
0FD8 7E JMP    0B54
0FDB E6 LDA B X00
0FDD 08 INX
0FDE FF STX    0BF4
0FE1 F7 STA B  0BF9
0FE4 4F CLR A
0FE5 5D TST B
0FE6 2A BPL    0FE9
0FE8 4A DEC A
0FE9 B7 STA A  0BF8
0FEC 18 XDX
0FED F3 ADD D  0BF8
0FF0 18 XDX
0FF1 BD JSR    0BA9
0FF4 FE LDX    0BF4
0FF7 20 BRA    0FBD
0FF9 01 NOP
0FFA 01 NOP
0FFB 01 NOP
0FFC 01 NOP
0FFD 01 NOP
0FFE 01 NOP
0FFF 01 NOP
0C00 4E 4F 50 NOP
0C03 4C 53 52 LSR
0C06 41 53 4C ASL
0C09 54 41 50 TAP
0C0C 54 50 41 TPA
0C0F 49 4E 58 INX
0C12 44 45 58 DEX
0C15 43 4C 56 CLV
0C18 53 45 56 SEV
0C1B 43 4C 43 CLC
0C1E 53 45 43 SEC
0C21 43 4C 49 CLI
0C24 53 45 49 SEI
0C27 53 42 41 SBA
0C2A 43 42 41 CBA
0C2D 54 41 42 TAB
0C30 54 42 41 TBA
0C33 58 44 58 XDX
0C36 44 41 41 DAA
0C39 53 4C 50 SLP
0C3C 41 42 41 ABA
0C3F 42 52 41 BRA
0C42 42 52 4E BRN
0C45 42 48 49 BHI
0C48 42 4C 53 BLS
0C4B 42 43 43 BCC
0C4E 42 43 53 BCS
0C51 42 4E 45 BNE
0C54 42 45 51 BEQ
0C57 42 56 43 BVC
0C5A 42 56 53 BVS
0C5D 42 50 4C BPL
0C60 42 4D 49 BMI
0C63 42 47 45 BGE
0C66 42 4C 54 BLT
0C69 42 47 54 BGT
0C6C 42 4C 45 BLE
0C6F 54 53 58 TSX
0C72 49 4E 53 INS
0C75 50 55 4C PUL
0C78 44 45 53 DES
0C7B 54 58 53 TXS
0C7E 50 53 48 PSH
0C81 52 54 53 RTS
0C84 41 42 58 ABX
0C87 52 54 49 RTI
0C8A 4D 55 4C MUL
0C8D 57 41 49 WAI
0C90 53 57 49 SWI
0C93 4E 45 47 NEG
0C96 43 4F 4D COM
0C99 52 4F 52 ROR
0C9C 41 53 52 ASR
0C9F 52 4F 4C ROL
0CA2 44 45 43 DEC
0CA5 49 4E 43 INC
0CA8 54 53 54 TST
0CAB 43 4C 52 CLR
0CAE 41 49 4D AIM
0CB1 4F 49 4D OIM
0CB4 45 49 4D EIM
0CB7 54 49 4D TIM
0CBA 4A 4D 50 JMP
0CBD 53 55 42 SUB
0CC0 43 4D 50 CMP
0CC3 53 42 43 SBC
0CC6 41 4E 44 AND
0CC9 42 49 54 BIT
0CCC 4C 44 41 LDA
0CCF 53 54 41 STA
0CD2 45 4F 52 EOR
0CD5 41 44 43 ADC
0CD8 4F 52 41 ORA
0CDB 41 44 44 ADD
0CDE 43 50 58 CPX
0CE1 42 53 52 BSR
0CE4 4A 53 52 JSR
0CE7 4C 44 53 LDS
0CEA 53 54 53 STS
0CED 4C 44 58 LDX
0CF0 53 54 58 STX
0CF3 2A 2A 2A ***
0CF6 00 00 00 ...
0CF9 00 00 00 ...
0CFC 00 00 00 ...
0CFF 00       .

0D00 51 01 00 01 Q...
0D04 51 01 51 01 Q.Q.
0D08 01 41 02 41 .A.A
0D0C 03 01 04 01 ....
0D10 05 01 06 01 ....
0D14 07 01 08 01 ....
0D18 09 01 0A 01 ....
0D1C 0B 01 0C 01 ....
0D20 0D 01 0E 01 ....
0D24 51 01 51 01 Q.Q.
0D28 51 01 51 01 Q.Q.
0D2C 0F 01 10 01 ....
0D30 11 01 12 01 ....
0D34 13 01 14 01 ....
0D38 51 01 51 01 Q.Q.
0D3C 51 01 51 01 Q.Q.
0D40 15 82 16 82 .T.T
0D44 17 82 18 82 .T.T
0D48 19 82 1A 82 .T.T
0D4C 1B 82 1C 82 .T.T
0D50 1D 82 1E 82 .T.T
0D54 1F 82 20 82 .T T
0D58 21 82 22 82 !T"T
0D5C 23 82 24 82 #T$T
0D60 25 01 26 01 %.&.
0D64 27 21 27 31 '!'1
0D68 28 01 29 01 (.).
0D6C 2A 21 2A 31 *!*1
0D70 27 51 2B 01 'Q+.
0D74 2C 01 2D 01 ,.-.
0D78 2A 51 2E 01 *Q..
0D7C 2F 01 30 01 /.0.
0D80 31 21 51 01 1!Q.
0D84 51 01 32 21 Q.2!
0D88 01 21 51 01 .!Q.
0D8C 33 21 34 21 3!4!
0D90 02 21 35 21 .!5!
0D94 36 21 51 01 6!Q.
0D98 37 21 38 21 7!8!
0D9C 51 01 39 21 Q.9!
0DA0 31 31 51 01 11Q.
0DA4 51 01 32 31 Q.21
0DA8 01 31 51 01 .1Q.
0DAC 33 31 34 31 3141
0DB0 02 31 35 31 .151
0DB4 36 31 51 01 61Q.
0DB8 37 31 38 31 7181
0DBC 51 01 39 31 Q.91
0DC0 31 0E 3A 0F 1.:.
0DC4 3B 0F 32 0E ;.2.
0DC8 01 0E 3C 0F ..<.
0DCC 33 0E 34 0E 3.4.
0DD0 02 0E 35 0E ..5.
0DD4 36 0E 3D 0F 6.=.
0DD8 37 0E 38 0E 7.8.
0DDC 3E 0E 39 0E >.9.
0DE0 31 03 3A 08 1.:.
0DE4 3B 0B 32 03 ;.2.
0DE8 01 03 3C 08 ..<.
0DEC 33 03 34 03 3.4.
0DF0 02 03 35 03 ..5.
```

'immediate mode' instructions, of which there are four: 'AIM' (And Immediate Mode), 'EIM' (Exclusive or), 'OIM' (Or) and'TIM' (Test). Each instruction requires one byte of immediate data, and a direct or indexed mode memory address. These instructions perform the appropriate logical operation on the immediate data and the contents of the specified location. The 'TIM' instruction is similar to the 'AIM' instruction, and not to the 6800 instruction 'TST'. The machine code requires that the immediate data is the first byte following the op-code, and the address the second byte.

The Basic program provided stores the machine code, held in data statements, and transfers control to the machine code program at &HOBOO. The program operates as follows:

1000 Prints 'wait' message.

1010 Stores machine code program.

1020 & 1030 Define functions to obtain the upper and lower bytes of an address entered as a string.

1040 Selects the language, sets top address for machine code stack and start address of machine code program.

1050 & 1055 Input & store the start address for disassembly.

1060 & 1065 Input and store the end address for disassembly.

1070 Executes machine code until a character needs to be printed, or disassembly is finished.

1080 Loops back to line 1040,

1081 Checks whether the user wishes to terminate the program following a null input in lines 1050 or 1060.

1090 to 1690 Machine code data.

Machine code program

The machine code program set up by Basic operates as follows:

0B00-0B1B Save the Basic stack pointer, load the machine code stack pointer and transfer control to the main body of machine code at 0F00 onwards.

Subroutines:

0B20-0B2B Saves machine code stack pointer, loads basic stack pointer, and returns to Basic.

0B30-0B3B Character input routine from the keyboard.

0B40-0B4E Character output routine to the display.

0B54-0B5C Routine to terminate the machine code program.

0B60-0B6A Inputs one hexadecimal digits.

0B6B-0B78 Inputs two hexadecimal digits.

0B80-0B8C Inputs four hexadecimal digits.

0B90-0B98 Prints one hexadecimal digit.

0B9A-0BA8 Prints two hexadecimal digits.

0BA9-0BB2 Prints four hexadecimal digits.

0BB8-0BBE Prints a space.

0BC0-0BC3 Sets output mode to LCD.

0BC4-0BCB Sets output mode to internal printer.

Main program:

0F00-0F26 Start of main body of disassembler, initialization, etc.

0F27-0F2E Prints carriage return and line feed.

0F31-0F40 Prints address of instruction and op-code.

0F43-0F64 Prints mnemonic.

0F67-0F87 Prints register, if any.

0F8A-0FA4 Prints mode character, if any, ie '#', '$' or 'X'.

0FA7-0FAD Jumps to 0FDB if branch relative/BSR.

0FAF-0FBB Outputs one or two bytes of either address or data.

0FBD-0FC3 Branches to 0FCE to terminate the program if disassembly is completed.

0FC5-0FCC Branches back to 0F27.

0FDB-0FF7 Calculates and prints the absolute address, for a relative branch or 'BSR', from the relative offset, and branches to 0FBD.

Locations 0C00-0EFF hold two tables of 0C00-0CF5 contains a list of mnemonics, and 0D00-0EFF contains two bytes per op-code, the first byte representing the mnemonic number, and the second byte the relevant details.

Using the program

On typing 'RUN' the program displays 'PLEASE WAIT: LOADING' to indicate that the machine code is being set up. This is followed by prompts for the start and end addresses of the memory area to be disassembled. The addresses are entered in hexadecimal but the prefix '&H' is not required. The program may be terminated at this point by pressing 'RETURN', and the user confirms that the program is to be terminated by entering 'Y', or 'N' to continue.

The program prints the disassembled program, one instruction per line, in five columns. Column one contains the instruction's address as a four digit hexadecimal number. The second column contains two hexadecimal digits representing the op-code, and third column gives the corresponding mnemonic. The fourth column gives the register, if applicable, and the fifth column contains the address or data field, in either immediate ('#'), direct ('$'), indexed ('X') or extended mode.

All branches and branch to subroutine instructions are followed by an absolute address.

Illegal op-codes are printed as '***', and the following mnemonics have been changed: 'INX', 'DEX', 'LDD' and 'STD' are printed as 'INC X', 'DEC X', 'LDA D' and 'STA D' respectively, and 'XGDX' has been changed to 'XDX'.