HANDEVELOP

HARDWARE & PROGRAMMING DESCRIPTION

Designed by HOWARD DELMAN

FEB 13, 1989	REV 1.0	Original draft
MAR 17, 1989	REV 1.1	Change name/Correct mistakes/ change port,SEGWR/cartridge access
APR 04, 1989	REV 1.2	Add cartridge information
APR 27, 1989	REV 2.0	Add TRAP & forced NOP info/correct breakpoint processing
MAY 15, 1989	REV 2.1	Minor changes
MAY 24, 1989	REV 3.0	New memory map/new CBUS/new analyzer functions
JUN 08,1989	REV 3.1	Minor changes

1.0 Overview:

Handevelop is a software development system for the Handy consumer game. Working with an Amiga computer, Handevelop allows a game programmer to load programs; control their execution; set breakpoints on address, data, control, and external matches; observe a trace of bus activity, including optional Suzy activity and video/refresh; and time arbitrary events. Numerous options allow great flexibility.

Handevelop contains the two custom chips used in Handy, as well as additional hardware for the debugging functions. Additionally, the DRAM used in Handy has been replaced by Static RAM.

In this document, all address are in hexadecimal.

Handevelop operates in one of two modes - execute and debug. A dip switch selects which mode is entered from reset. If entry into debug mode is selected, an NMI is forced by the hardware immediately after reset negates.

A memory map of Handevelop is given in Table 1. All named addresses are typed in capital letters. For multi-byte reads or writes, the most significant byte of data is at the least significant address, maintaining the standard of the 6502.

1.1 Execute mode:

In execute mode, an application program is being run. The debugger is completely transparent, and operation proceeds exactly as it would in a real Handy.

1.2 Debug mode:

In debug mode, the CPU is either executing special debug code or is waiting for further instructions. When in debug mode, part of Handy's address space is mapped away for use by Handevelop. Specifically, the space from 0000 to 3FFF will not decode to Handy RAM, but to Handevelop RAM, ROM, and other registers. A control bit in Handevelop allows a remapping of Handy RAM to make the masked area available. By setting the bit SEG, the RAM from 0000 to 3FFF will be remapped to 4000 to 7FFF. The RAM normally at those locations will be masked. With SEG cleared, the RAM is normally mapped, with 4000 to 7FFF appearing normally and 0000 to 3FFF masked. Of course, in execute mode, all Handy RAM appears normally, and SEG has no effect. This is summarized in Table 3.

TABLE 1
HANDEVELOP MEMORY MAP

0000-07FF	RAM	R/W	D0-D7	
1000-1001	RCARTEN	R/W	D0-D7	Read and write cartridge
1100-11FF	ABREN	R/W	D0-D3	Set Abus breaks; Dn -> Channel n
1200-12FF	DBREN	R/W	D0-D3	Set Dbus breaks; Dn -> Channel n
1300-13FF	CBREN	R/W	D0-D3	Set Cbus breaks; Dn -> Channel n
1400-1403	ANALRD	R	D0-D7	32 bits of trace data
1510-1512	ANADD	R/W	D0-D7	24 bits of current trace address
1518-151A	FRSTBRK	R	D0-D7	24 bits of first break address
1520-1523	TIMERDEN	R	D0-D7	32 bits of timer value
1540-1542	LDDEL0	W	D0-D7	24 bits of channel 0 break delay
1543	BRKCON0	W	D0-D3	Any write clears BRKL0
	AEN0	W	D0	0 -> Disregard Abus 1 -> Enable Abus compares
	DEN0	W	D1	0 -> Disregard Dbus 1 -> Enable Dbus compares
	CEN0	W	D2	0 -> Disregard Cbus 1 -> Enable Cbus compares
	DELSEL0	W	D3	0 -> Clock delay with PH2 1 -> Clock delay with Sync
1550-1552	LDDEL1	W	D0-D7	24 bits of channel 1 break delay
1553	BRKCON1	W	D0-D3	Any write clears BRKL1
	AEN1	W	D0	0 -> Disregard Abus 1 -> Enable Abus compares
	DEN1	W	D1	0 -> Disregard Dbus 1 -> Enable Dbus compares
	CEN1	W	D2	0 -> Disregard Cbus 1 -> Enable Cbus compares
	DELSEL1	W	D3	0 -> Clock delay with PH2 1 -> Clock delay with Sync
1560-1562	LDDEL2	W	D0-D7	24 bits of channel 2 break delay
1563	BRKCON2	W	D0-D3	Any write clears BRKL2
	AEN2	W	D0	0 -> Disregard Abus 1 -> Enable Abus compares
	DEN2	W	D1	0 -> Disregard Dbus 1 -> Enable Dbus compares
	CEN2	W	D2	0 -> Disregard Cbus 1 -> Enable Cbus compares

	DELSEL2	W	D3	0 -> Clock delay with PH2 1 -> Clock delay with Sync
1570-1572	LDDEL3	W	D0-D7	24 bits of channel 3 break delay
1573	BRKCON3	W	D0-D3	Any write clears BRKL3
	AEN3	W	D0	0 -> Disregard Abus 1 -> Enable Abus compares
	DEN3	W	D1	0 -> Disregard Dbus 1 -> Enable Dbus compares
	CEN3	W	D2	0 -> Disregard Cbus 1 -> Enable Cbus compares
	DELSEL3	W	D3	0 -> Clock delay with PH2 1 -> Clock delay with Sync
1580-1581	PORT	R/W	D0-D7	See TABLE 2
1590	TMCON	W	D0-D1	Sel. clock for timer 00 -> 16MHz 01 -> PH2 10 -> SYNC 11 -> CL2
15A0	ANCON	W	D0-D5	
	SUZEN	W	D0	0 -> Do not trace Suzy cycles 1 -> Trace Suzy cycles
	VREN	W	D1	0 -> Do not trace Vid/Ref cycles 1 -> Trace Video/Refresh cycles
	~~DATEN~~	W	D2	0 -> Do not trace Data cycles 1 -> Trace Data cycles
	~~OPEN~~	W	D3	0 -> Do not trace Opcode cycles 1 -> Trace Opcode cycles
	STOPEN	W	D4	0 -> Forbid analyzer disables 1 -> BRK1 may disable analyzer
	STARTANAL	W	D5	0 -> Disable analyzer 1 -> Enable analyzer
15C0	LABRK	W	D0-D7	Upper byte of Abus RAM address
15D0-15D2	LDRADD	W	D0-D7	20 bits of cartridge address
15D3	ROMPAGE	W	D0-D5	
			D0-D2	Primary memory paging
			D3-D5	Secondary memory paging 000 -> 4096 bytes/page 001 -> 2048 bytes/page 010 -> 1024 bytes/page 011 -> 512 bytes/page 100 -> 256 bytes/page
15E0	SEGWR	W	D0	0 -> No remapping of Handy RAM 1 -> Remap 4000-7FFF in Handy RAM
15F0	TRAPCLR	W		Clear trap vector

1600	BRKRD	R	D0-D7	
	BRKLn	R	Dn	0 -> No break on channel n 1 -> Break occurred on channel n
	TRAP	R	D4-D7	Trap number
1601	INTDLY	R	D0-D2	No. of extra cycles added to PC
3000-37FF	ROM	R	D0-D7	

TABLE 2

PORT OPERATION

1000	STATUS	R	D6-D7	BUSY -> D6 DATA AVAILABLE -> D7
1001	DATA	R/W	D0-D7	PORT DATA
1002	DIRECTION	W	D7	0 -> OUTPUT 1 -> INPUT
1003	POUT	W	D7	D7 -> POUT

NOTE: When receiving data, DATA AVAILABLE goes HI when data has been received. When sending data, DATA AVAILABLE goes LOW until the Amiga acknowledges the transfer.

TABLE 3

HANDY RAM MAPPING

MODE	SEG	CPU ADDRESS	HANDY RAM ADDRESS
EXECUTE	X	0000 - FFFF	0000 - FFFF
DEBUG	0	0000 - 3FFF	no select
		4000 - 7FFF	4000 - 7FFF
		8000 - BFFF	8000 - BFFF
		C000 - FFFF	C000 - FFFF
DEBUG	1	0000 - 3FFF	no select
		4000 - 7FFF	0000 - 3FFF
		8000 - BFFF	8000 - BFFF
		C000 - FFFF	C000 - FFFF

2.0 Operation:

The transition to debug mode is brought about by a non-maskable interrupt. This interrupt can be generated by either the breakpoint circuity, a software trap, or the Amiga computer. The transition to execution mode occurs whenever an RTI instruction is encountered while in debug mode. These transitions make no change to Handy, and its stack is not disturbed.

2.1 Handevelop Control:

The Handevelop hardware resides in 16K bytes of address space starting at address 0000. Within this space is 2K bytes of Handevelop ROM and 2K bytes of Handevelop RAM. A communication port provides interfacing between Handevelop and the Amiga computer.

Additionally, numerous registers are located in the address space. They will each be described in appropriate places in this document.

2.2 Breakpoints:

Handevelop contains four identical breakpoint circuits, referred to as channels 0 - 3. Each channel can provide many break combinations. Breakpoints can be set to test the 16 bit address bus (Abus), the 8 bit data bus (Dbus), a 4 bit external bus, and the three control lines R/W, Cyc1 and Cyc0. The two Cyc signals indicate what type of bus tansfer is occuring. The four external bits and the three control bits are grouped together in the hardware and are referred to collectively as the Cbus, a seven bit bus. The arrangement of the Cbus and types of cycles are given in Table 3.

Each channel has a control register, BRKCONn, containing a bit to independently enable each bus. If none of the three buses are enabled for a channel, the channel is turned off. A channel only looks for bus matches with enabled buses.

In the hardware, each bus has a RAM. There is a RAM address for every possible bus value. The Abus RAM has 2^{16} locations, the Dbus RAM 2^8, and the Cbus RAM 2^7. Each RAM is four bits wide, one bit for each channel. The RAMs are readable and writeable by the CPU when in debug mode. In order to cause a break at a particular bus value, a ONE must be stored at the address that corresponds to that value, in the bit position corresponding to the desired channel. Examples are given in Appendix A.

Locations in the Dbus RAM and Cbus RAM are accessible directly as an index off of DBREN and CBREN respectively.

The Abus RAM, due to its size, needs to be accessed indirectly. The RAM is partitioned into 256 pages of 256 bytes each. The selected page, representing the upper eight bits of the desired value, is stored at location LDABRK. The contents of the page, representing the lower eight bits of the desired value, is accessed as an index off of ABREN.

When a break occurs, the NMI may not occur immediately. The user has the ability to set a delay before interrupting. This delay is determined by a 24 bit value stored at three locations starting st LDDELn. The delay counter can be programed to count either CPU cycles, or op-code fetches. This selection is accomplished by another bit in BRKCONn.

To determine which channel caused the break, and to see if any other breaks are pending (waiting for the delay counters to count down), read BRKRD. Each channel has a bit which is set when a break occurs. The bit is cleared when a write occurs to that channel's BRKCON.

A delay of zero will cause the interrupt at the break. In general, the instruction causing the break will complete, and the CPU will attempt to execute one or more additional instructions. Handevel forces NOP instructions on the data bus during this transition period, so that the only change to the system after the break is to the CPU program counter. By reading INTDLY, the number of NOPs forced can be read. The PC should be adjusted back by this amount.

TABLE 4

CBUS ORGANIZATION

BIT POSITION	SIGNAL	FUNCTION
0	EXT 0	EXTERNAL PROBE
1	EXT 1	EXTERNAL PROBE
2	EXT 2	EXTERNAL PROBE
3	EXT 3	EXTERNAL PROBE
5,4	CYC1,CYC0	00 -> SUZY CYCLE 01 -> VIDEO/REFRESH CYCLE 10 -> CPU DATA CYCLE 11 -> OP-CODE FETCH
6	WE/	0 -> WRITE CYCLE 1 -> READ CYCLE
7	BRK	0 -> NO BREAKS ON THIS CYCLE 1 -> A BREAK OCCURRED

NOTE: Bit 7 only goes to the analyzer display. Bit 7 of the Cbus RAM is always zero

2.3 Trap Generator:

The trap generator allows the setting of software breakpoints. In normal operation, an opcode ending in 3 (03,13 .. F3) is a NOP. The trap generator will generate an interupt any time one of these opcodes is encountered, except 03. The upper nibble of the opcode is saved as the trap number, and may be read as the upper nibble of BRKRD. The trap number will be cleared by any write to TRAPCLR.

Traps are logically identical to a hardware breakpoint with DBUS = X3, CBUS = OP-CODE, Delay = 0. Traps affect the first break register, as well as the break bit in trace memory.

2.4 Analyzer:

The analyzer consists of a circular buffer, 32 bits wide. The size of the buffer is determined by the amount of RAM installed on the Handevelop board. The minimum amount is 32K bytes, and the maximum is 128K bytes.

Generally, the analyzer records the values on the three buses, as well as the logical OR of the outputs of the four breakpoint channels and the trap generator. There are, however, optional modes of operation, controlled by bits in ANCON.

There is a bit in ANCON corresponding to each type of bus cycle. At reset, all four bits are set and all bus activity will be traced. Clearing selective bits, suppresses tracing of that type of cycle. Thus, one could, for example, trace only op-code fetches, or only Suzy activity. All possible combinations are valid.

If the bit STOPEN is set, the analyzer will stop recording when BRK1 occurs. It will restart when BRK2 occurs. It can be manually restarted by Handebug by setting the STARTANAL bit in ANCON. This will assure tracing immediately upon returning to Handy mode. The analyzer can be manually stopped by Handebug by clearing the STARTANAL bit in ANCON. Tracing will not begin until a BRK2 occurs.

Locations in the trace buffer are referenced by a 24 bit address. Some of the upper bits may not be relevant, depending on the amount of trace RAM. In this case, these extra bits are truly "don't care". They will count, and may be read and written, but they have no effect on trace operation.

The trace RAM may be read indirectly, by reading from four locations starting at ANALRD. The trace address was autoincrementing as data was being recorded, and it

autodecrements as data is read from trace RAM. The decrement occurs after reading the last byte (ANALRD+3) of a trace. The first trace read will be irrelevant and should be discarded. The second trace will be the last one recorded before the interrupt. This is due to the fact that both increment and decrement are post-cycle.

The present trace address may be read at three locations starting at ANADDRD. When the first break or trap occurs after leaving debug mode, the trace address is saved and may be read at three locations starting at FBRKADD. The trace address may be set to any value by writing to three locations starting at ANALLD. When returning to execute mode, tracing will begin at the last value in ANADDRD.

2.5 Timer:

The timer is a 32 bit counter. It begins timing when a BRK3 occurs. It stops when any other break occurs. Subsequent BRK3s will reset and restart the timer. It cannot be stopped and started. The timer does not run in debug mode.

There are four user selectable clock options for the timer: 16MHz, PH2, Sync, and CL2. 16MHz allows for real-time timing. PH2 and Sync are useful for software analysis. CL2 allows determining which scan line is active. The choice of clock is determined by two bits TMESEL0 & TMESEL1.

The Handevelop timer may be read by reading four locations starting at TIMERDEN.

3.0 Emulation Memory:

The normal Handy cartridge space is available on Handevelop. The sockets on the Handevelop PCB will support either RAM or EPROM, and either 256K or 1Meg parts may be used. Only 1Meg parts will provide the full 1 Megabyte of address space.

In Handy mode, the memory acts exactly as a Handy cartridge would. Handevelop provides a means to select the memory page size. By storing a value in ROMPAGE, the page size may be set to 256, 512, 1024, 2048, or 4096 bytes. A game program that over-reads its page size will wrap around correctly (or incorrectly, depending on your point of view) to address zero.

The primary and secondary cartridge memory sections may be individually paged. Each may have its own choice of memory type. Note that memory type (RAM, ROM, and size) must be the same within a memory section.

In debug mode, it is not necessary to use the Handy shifting and paging schemes. A twenty bit register points to any location in cartridge space. The register is auto-post-increment when reading and writing. The register is loaded by storing two and a half bytes, starting at LDRADD.

Before accessing memory through this register, the page size should be set to 4096 (full - size). Information can now be accessed contiguously throughout memory without concern for paging. Be sure and restore the correct page data when finished. Primary memory is accessed through location RCARTEN. Secondary memory is accessed at RCARTEN+1. Both sections use the same address register.

APPENDIX A
BREAKPOINT EXAMPLES

EXAMPLE 1: Break on any access of address 6000, no delay. Use channel 0.

```
Set address break bit:   Write 60 to LABRK
                         Read ABREN+0
                         Logical OR 01
                         Store ABREN+0
Set control:             Store 01 to BRKCON0
Set delay:               Store 00 to LDDEL0,LDDEL0+1,
                                     LDDEL0+2
```

EXAMPLE 2: As above, but break only on writes.

```
Set control RAM:         Load 00 in X
                         Read CBREN,X
                         Logical OR 01
                         Store CBREN,X
                         Inc X
                         Repeat if X < 40

Set control:             Store 05 to BRKCON0
```

EXAMPLE 3: As above, but delay 1080x instructions.

```
Set delay:               Store 80 to LDDEL0
                         Store 10 to LDDEL0+1
                         Store 00 to LDDEL0+2

Set control:             Store 0D to BRKCON0
```

EXAMPLE 4: As above, but also break on any Suzy read in the address range 2000 - 20FF. Use channel 1.

```
Set address bits:        Write 20 to LABRK
                         Load 00 in X
                         Read ABREN,X
                         Logical OR 02
                         Store ABREN,X
                         Inc X
                         Repeat unless X = 0

Set control RAM:         For locations eq. to XX00XXXb
                         Read CBREN,X
                         Logical OR 02
                         Store CBREN,X Set control:

Set control:             Store 05 to BRKCON1

Set delay:               Store 00 to LDDEL1,LDDEL1+1,
                                     LDDEL1+2
```