

SBC-386EX-2

Guide to the Board Schematic

Page 1:

This is the index to the other pages in the Kicad schematic. The only components are the Spare gates and the pull-up resistors used in various places across the schematic hierarchy.

Page 2: CPU386EX

The CPU part is the Adapter board for the CPU. It has 2 extra VCC and 2 extra GND pins. Otherwise, it is the same as the CPU part. The FPU is strapped to run at the same frequency as the CPU. Only the Intel FPU's can run at a different frequency, but in such a mode do not power-down when not in use. The Cyrix FPU, and possibly other brands, only run at the CPU frequency.

RESET_0# is used for the TRST# input so that the test-pin/boundary scan reset arrives earlier than the CPU reset. This worked fine on the prototype board.

UCS# is connected to BS8# as signal CS_ROM#. This is the only select that must force 8-bit accesses; all other 8-bit access, memory or I/O, is specifically programmed in the chip-select registers.

Board 2.0 / dual: the Adapter part (U1) and the 386EX CPU (U101) are overlaid on the schematic. The connections on the PC board itself are manually created.

Page 3: Static-RAM

Some future board may eliminate static RAM, but it proved so useful on the KISS-68030 for DRAM testing, that provision for a 32Kbyte chip is provided. ROM is also on this page, and with a jumper, 64Kbyte EPROM, or Flash 128K, 256K, or 512K may be used. 128Kbyte SST flash is suggested for board debug.

The "M386EX2" GAL16V8 excludes SRAM and external ECB memory accesses from the DRAM area. It also decodes the IDE chip selects, and reverses the read and write signals for fly-by DMA access. The ECB data access is only enabled on external I/O (XIO#) or external memory (XMEM#) access.

Page 4: UART-SIO

Serial I/O at \$3F8 using the SIO0 peripheral on the 386EX chip is converted to RS-232 level signals. The IBM 9-pin serial port is wired so that RTS from the UART is directed to pins RTS and DTR on the connector. CTS input to the UART is jumper selected as connector CTS or DSR. Modem protocol using RTS/CTS can thus be used with terminals requiring either RTS/CTS or DTR/DSR control protocols.

Page 5: RTC_LITES + (reset, & run/halt)

The DS1302 RTC+NVRAM chip is connected to the 386EX chip using only 3 pins. Data in/out, “DQ” must be set up as an open-collector drive, and set to ‘1’ to read input. Pins are different from the prototype board, but the programming should be identical.

The Reset circuit is common to many RetroBrew boards, and has the same jumper to select SBC v1 Legacy or Kontron interpret of the ECB B_RESET (C-31) signal. The LITES, 4 LED outputs, are the same as the prototype board, and are connected to P1.4-7. A programmed ‘0’ will light a LED, and a programmed ‘1’ will extinguish it. The Run/Halt bi-color LED is driven from the LS240 on page 6. R1, specified on the schematic as 68 ohms, should be in a socket. The value may be adjusted up or down to select the desired LED brightness. If driven from a 74ACT240, instead of a 74LS240, then a higher resistance should definitely be used.

The resistor is part of the resistor array on page 10, and a machine-tool 14-pin socket is suggested for those components.

Page 6: ECB drivers

The chips which connect signals to the ECB bus are on this page. Certain chips may be omitted if not using some aspects of the ECB bus; viz.,

If not using ECB memory accesses (VGA), then U12 & U13 may be omitted.

If not using ECB memory or ECB I/O, then U10 & U11 may be omitted also.

If not using the ECB bus in any way, and no FPU is present, then U3 may be omitted. With the FPU, U3 is required.

U14 is required whether using the ECB bus or not. It holds the IRx chip inputs low, even if not used.

Page 7: ECB bus

These are the connection to to the 96-pin DIN connector. All 41 design rules checks in Kicad occur on this page because 41 of the global labels do not appear anywhere else.

Decoupling caps and the stand alone power connector are specified. Note that not all of the 0.1uF caps are used on the PCB. The two 22uF caps may be replaced by larger value caps if desired. (I probably will.)

DMA0 is connected to the ECB bus for 2-cycle transfers. Fly-by transfers are not an option. B_TEND# is driven as EOP# from the DMA controller is an experimental connection.

Bus Int	386EX	ICU IRQ	
n/c	n/a	IRQ0	fixed Timer 0 instrrupt (18.2hz)
IR1 & INT (IR2)	INT0	IRQ1	should be Keyboard
(IR3)	n/a	(IRQ2)	redirected from IRQ9, the cascade input
n/c	n/a	(IRQ3)	redirected from IRQ13 (INT8 pin is used as P3.1)
IR5	SIO0	IRQ4	COM1 interrupt
IR6	INT1	IRQ5	
IR7	INT2	IRQ6	
n/c	INT3	IRQ7	
n/c	INT4	n/c	pin used as Timer 0 clock input

IR2	INT5	IRQ9	which is re-directed to IRQ2
n/c	n/a	IRQ10	fixed Timer 1 interrupt
n/c	n/a	IRQ11	fixed Timer 2 interrupt
n/c	n/a	IRQ12	DMA interrupt (programmable)
IR3	INT6	IRQ13	which is re-directed to IRQ3
n/c	INT7	IRQ14	IDE interrupt
n/c	n/a	IRQ15	watchdog timer

Page 8: Oscillators

U16 is the CPU double frequency input. The fastest N80386EX chips are 33mhz, so 66mhz is specified. For accurate OS timing, use an oscillator with a frequency that is an EVEN INTEGER. This will affect the timing of the PSCLK used as a Timer input at exactly 1.000mhz. Using another value oscillator will put the timing off by a small percentage; e.g., a 66.667mhz oscillator will result in a 1% timing error. So don't sweat it.

The UART is driven from a standard 1.8432mhz oscillator for MSDOS compatibility. The frequency is divided by 2 for input to Timer0, which is used to create the IBM 54.925 ms (18.206 hz) clock interrupt, and is the frequency standard for dynamically determining the CPU frequency. Use exactly 1.8432mhz.

Oscillators may be full can or half can.

Page 9: DRAM

DRAM is specified as 60 ns, 5 volt, 72-pin, single or double sided SIMM, CBR refresh, FPM or EDO (latter is preferred), parity or non-parity.

Single sided DRAM comes in sizes of 4Mb, 16Mb, and 64Mb. Double sided DRAM comes in sizes of 8Mb and 32Mb. DRAM size is determined by the correct setting of the PRD1 and PRD2 short-circuit resistors on the SIMM, and are interpreted within the DRAM-2 GAL.

The DRAM controller is contained within two GAL16V8 chips. GAL DCTL16-2 generates the RAS/CAS timing, and DRAM-2 determines 8/16-bit access, and drives the appropriate SIMM inputs. DRAM refresh uses CAS-before-RAS, aka CBR. The GAL chips should be 7ns or faster. A 16mhz CPU may use slower ones, and may use a slower SIMM (80ns).

Address multiplexing supports DRAM sizes from 4-64Mb. 74F257 chips are specified for speed, but for lower power consumption I want to try 74ACT257/157, and below 33mhz, even 74ALS157. Usage of the chips is such that 74xx257 or 74xx157 are both acceptable (F257 is a tad faster than F157).

Auxiliary use of the DCTL-2 GAL is to decode the CPU Halt state. (There was an extra input, and an extra output to do the decode here.)

Page 10: SD card

Due to space limitations, the suggestion to use a MicroSD card socket was adopted. This storage medium is entirely experimental. The 3.3v voltage supply is the same as other RetroBrew boards, and the 5v CMOS signal voltage dividers are the same as the Z180 Mark IV. Synchronous output on the 386EX is, however, done with 16-bit words, so programming may be a bit tricky. Only one clock output from the 386EX may be used, since

the two clocks are connected together; one is output, the other is input. At the moment I think the Transmit clock should be the timing source.

There is no Write Protect for the MicroSD cards. The socket supports Card Detect, which can be sensed at P3.1. Chip Select is derived at P3.6, an open-collector driver (watch out for 5v vs. 3.3v) which also drives the activity LED. The 270 ohm resistor should be in a socket, in case it loads the P3.6 output. Or it may be pulled if the LED load is too great.

The MicroSD card footprint is for the Molex 47309 series of sockets. They come in various heights, and I conjecture that the higher sockets may allow more room for SMD soldering. Both Mouser and Digikey have these sockets. Choose for soldering ease, not price. (More on this later).

Either DMA controller may be connected to the SSIO if desired.

Programming this beast is going to be a challenge.

Page 11: bus-IDE

The IDE connector is made to look as close as possible to one in a PC/AT. Programming the controller is through 8-bit I/O operations, but data transfer is set up for 16-bit DMA only. Fly-by transfers are intended. If other than fly-by DMA transfers are to be used (meaning 2-cycle DMA), then the DACK1# signal must be defeated within the 386EX chip.

The board is laid out for a right-angle, non-shrouded, 40-pin connector array. This puts the plane of an IDE-to-CF card adapter in the plane of the CPU board with the activity LED facing forward. Space is provided for a straight-through shrouded header, if desired. The shrouded header might be preferred on a board that is not to be plugged into an ECB backplane.

Epilogue:

The Intel 386EX chip is an amazing piece of peripheral integration. And this board is one h**1 of a big project.

2018-Mar-10 "Page 7 / ECB bus" interrupts added

2018-Jan-21

2018-Jan-17

John Coffman