Excerpt from “GIMIX 6809+ CPU BOARD” manual –

EXTENDED ADDRESSING

The 6809 CPU is capable of addressing a maximum of 64K of memory space directly. This capability, while sufficient for many applications, is a limiting factor when larger, especially multi-user/multi-tasking, systems are considered. In order to expand the addressing capability of the 6809 the SS5OC bus definition includes 4 extra or extended address lines. These additional lines give the bus a total of 20 address lines and allow for up to 1 MBYTE of address space. Since the 6809 generates only 16 address lines, some combination of hardware and software must be used to simulate the extra 4 signals. NOTE: In order to take advantage of extended addressing the devices to be addressed must be capable of decoding all 20 address lines. Boards such as the GIMIX 32K STATIC RAM board, PORT SERIAL I/O board, and 128K PROM/ROM/RAM board have the capability to decode all 20 address lines and can be used with extended addressing. The GIMIX 6809 CPU also provides extended address decoding for the 8 devices decoded by the FPLA.

The GIMIX 6809 CPU has provisions for three methods of simulating the 4 extra address lines. These methods include STRAIGHT BANK SELECT, which uses a simple latch, and two different methods of Dynamic Address Translation (DAT) which can also generate the 4 extended address signals. Each method has advantages and disadvantages and it is up to the user to determine which method best suits his needs. In addition, some software available from various vendors may require that a particular method be used.

STRAIGHT BANK SELECT

In the straight bank select method of extended addressing the 1 MBYTE of available address space is divided into 16 sections or banks of 64K each. Any devices in the system that are set up to decode a particular bank address will only appear on the bus when their bank address appears on the extended address lines of the bus. For example if a multi-user system were being set up in which each of three users was to have 32K of memory dedicated to his use, three 32K memory boards could be installed in the system . Each of the 3 boards could be set to the same base addresses $0000-$7FFF but with each set to a different bank. When a particular user was to have access to his portion of memory the monitor/supervisor program would place the bank address of that user on the extended address bus and his bank of memory would be enabled. Areas of memory and devices such as I/O, disk controllers etc., that were to be shared by all users, would be set to ignore the extended address lines and would appear at the same address in all banks. While this method is relatively simple and straightforward it has limitations. For example it requires that each user or task be permanently assigned a particular amount of dedicated memory. If a user needs more memory than he is originally assigned the hardware must be reconfigured to increase his available memory. If a user does not require all of the memory allocated to him the unused portion is unavailable to other users unless the hardware configuration is changed. Also since it may only be possible to assign an entire board to a particular bank, memory may be wasted because a full 32K must be assigned to a user who only requires 16K.

BANK SELECT LATCH

When using the STRAIGHT BANK SELECT method of extended addressing, the data on the extended address lines is determined by the value stored in the BANK SELECT LATCH.

The bank select latch is a write-only device that appears at memory location $FFFF. A processor write to this location stores data in the latch. A read from this location returns data from any other device addressed at this location. Normally a read from this location would return the restart vector stored in a PROM/ROM monitor located from $FFFF down. The least significant 4 bits written to the latch are the bank number $0-$F of the bank to be enabled.

Any software that modifies the contents of the bank select latch must be located in an area of memory that is not affected by switching banks. In most applications bank switching would be done by a monitor/supervisor program located in memory shared by all users and not affected by the extended address lines.

Note: Two of the remaining bits of the latch are used to control other functions on the board. If either of these functions are in use any programs that modify the bank select latch must not inadvertently change these two bits. Since the bank select latch is a write-only device, the last data written to the latch should be kept in temporary storage for comparison when new data is to be written. Bit 5 of the bank select latch is the software control latch for the FPLA address decoder and bit 4 controls the user defined latch output at solder pad "A". See the appropriate section of the manual for further information on these features.

DYNAMIC ADDRESS TRANSLATION (DAT)

Dynamic address translation is a method of memory management that allows better utilization of the memory resources within a system. It overcomes some of the disadvantages of the straight bank select method of extended addressing in that it allows available memory to be allocated among users as their requirements vary. It also allows memory boards that can only be addressed as large contiguous blocks to be effectively split into 4K segments that can be addressed as required. Since this re-location of memory is under software control it can be done at any time and does not require hardware changes in the system. DAT breaks the entire memory space into 4K segments, so memory assignment can more closely fit the requirements of each user/task. DAT has several other advantages, for example, the system monitor/ supervisor program could test each 4K segment of memory before assigning it to a user or task. If a bad segment of memory were found it could be eliminated from the table of available memory and a good segment substituted. The monitor/supervisor program could then set a flag or indicator to show that the system requires maintenance.

In order to understand how dynamic address translation works we must first understand the concept of physical and logical addresses. A physical address, as the name implies, is the address at which a device (memory, I/O, etc.) is set by its hardware to respond. For example the DIP switches on a memory board determine its physical address. A logical address is the address that the processor outputs when it attempts to access a particular memory location. For example when the processor attempts to write data to location $0000, it outputs the logical address $0000 on its address lines. In a system that does not have dynamic address translation, the address lines from the processor are connected directly to the bus and the physical and logical address are always equal. When the processor writes to location $0000 the memory at $0000 responds. Dynamic address translation allows us to assign a different logical address to memory that has a particular physical address. For example memory with a physical address of $0000 might be assigned the logical address $2000. When the processor attempts to access memory at logical address $2000 the memory at physical address $0000 would respond. DAT then is a method of translating from the processor to a physical address that appears on the bus.

DAT is implemented on the GIMIX 6809 CPU board by inserting a high speed, random access memory (the DAT RAM) between the upper 4 address lines of the 6809 and their corresponding bus buffers. A second identical RAM is used to generate the 4 bits of the extended address. The 4 address lines from the processor are used to address 1 of 16 locations in the DAT RAM and the data stored in these locations becomes the physical address that appears on the bus. The DAT RAM translates the upper 4 bits of the processor's logical address into the upper 4 bits of a physical address and 4 bits of extended address. Since the DAT only translates the upper 4 bits of address it divides the memory space into 16 4K logical segments. The physical address space consists of 16 4K physical address segments in each of 16 possible banks or a total of 256 possible 4K physical address segments. By placing the proper data in the DAT RAM any of the 256 physical segments can appear at any of the 16 logical address segments. For example 4K of memory physically addressed at $0000-$OFFF could appear to the processor to be addressed at $C000-$CFFF. A physical address segment could also be made to appear at more than one logical address. For example a 4K segment physically addressed at $EOOO-$EFFF could appear at $E000-$EFFF and $0000-$CFFF at the same time

.

NOTE: ON POWER UP THE DATA I N THE DAT RAM AND THEREFORE LOGICAL ADDRESSES ARE UNDEFINED. ONLY THE UPPER 256 BYTES OF MEMORY $FFOO-$FFFF ARE GUARANTEED TO BE AT THEIR PROPER ADDRESS IN SYSTEMS THAT USE DYNAMIC ADDRESS TRANSLATION THESE BYTES MUST CONTAIN, IN ADDITION TO THE RESET AND INTERUPT VECTORS, SOFTWARE THAT INITIALIZES THE DAT RAMS TO A PREDETERMINED STARTING CONFIGURATION. THIS INITIALIZATION MUST TAKE PLACE BEFORE ANY MEMORY ACCESSES OUTSIDE THIS 256 BYTE AREA ARE MADE.

For the purpose of writing data to the DAT, the DAT RAM shares the upper 16 bytes of address space $FFFO-$FFFF with the memory normally at these addresses (usually the system monitor PROM/ROM). The DAT RAM is write-only; a write to these locations stores data in the DAT RAM, a read from them returns data from the PROM/ROM monitor. The least significant 4 bits of data written to the DAT are stored in the RAM that translates the upper 4 address lines of the 6809, the upper 4 bits are stored in the RAM that generates the 4 bits of extended address.

Two different methods of dynamic address translation are available as options on the GIMIX 6809 CPU board. One is compatible with the method used on the SWTP MP-09 board and the SWTP SBUG-E monitor ROM. When the board is configured for this DAT method the SWTP SBUG-E ROM can be plugged directly into the GIMIX 6809 CPU board. The GIMIX 6809 can be used as a direct replacement for the SWTP MP-09 board in systems where baud rates are available from a source other than the CPU board. The GIMIX 6809 CPU does not have an on-board baud rate generator. The second DAT method is an enhanced version that allows much faster operation when switching tasks.

While dynamic address translation may find some use in smaller systems (64K and under), it will probably be most useful in larger multi-user, multi-tasking applications. For example, in the system described in the section on straight bank select, with 3 users in a multi-user application, each user (task) could be assigned the memory required from any of the available memory in the system. As the users' memory requirements changed, memory could be allocated or deallocated as necessary. With the SWTP compatible DAT, switching users (tasks) requires writing a new set of 16 values into the DAT RAM each time the system switches between users (tasks). With the GIMIX enhanced DAT method the DAT values for up to 16 different users (tasks) are written to the DAT RAM and switching between users is done by writing a single byte to a task select resister. Each time the system switches users (tasks) only the task select resister byte need be written.

SWTP COMPATIBLE DAT

To use this version of DAT the user must write values to 16 locations in the DAT RAM. Each of the 16 locations corresponds to one of the 16 possible 4K logical address segments. The first location $FFFO corresponds to logical address segment $0000-$0FFF the second $FFF1 to $1000-$1FFF and so on up to the 16th at $FFFF which corresponds to logical address segment $F000-$FFFF. The least significant 4 bits of data written are the COMPLEMENT of the upper 4 bits of the desired physical address. The upper 4 bits are the desired bank address. For example if a 4K segment of memory located at physical address $2000 in bank 1 is to appear at logical address $0000 then the value $1D would be written to the DAT RAM at location $FFFO. The 1 indicates that the desired segment is physically located in bank 1 and the D is the complement of the upper 4 bits of its physical address, $2000. The following table shows the 16 logical address segments with their corresponding DAT locations, sample data and the resulting physical addresses:

LOGICAL ADDRESS	DAT LOCATION	DATA	PHYSICAL ADDRESS	
			BANK	SEGMENT
$0000-$0FFF	$FFF0	$0F	0	
$1000-$1FFF	$FFF1	$0E	0	$1000-$1FFF
$2000-$2FFF	$FFF2	$0D	0	$2000-$2FFF
$3000-$3FFF	$FFF3	$0C	0	$3000-$3FFF

$4000-$4FFF	$FFF4	$1B	1	$4000-$4FFF
$5000-$5FFF	$FFF5	$09	0	$6000-$6FFF
$6000-$6FFF	$FFF6	$0A	0	$5000-$5FFF
$7000-$7FFF	$FFF7	$08	0	$7000-$7FFF
$8000-$8FFF	$FFF3	$07	0	$3000-$3FFF
$9000-$9FFF	$FFF9	$06	0	$9000-$9FFF
$A000-$AFFF	$FFFA	$05	0	$A000-$AFFF
$B000-$BFFF	$FFFB	$04	0	$D000-$£:FFF
$0000-$CFFF	$FFFC	$03	0	$C000-$CFFF
$D000-$DFFF	$FFFD	$02	0	$D000--$DFFF
$E000-$EFFF	$FFFE	$01	0	$E000-$EFFF
$F000-$FFFF	$FFFF	$00	0	$F000-$FFFF

Notice that in the above table:

1 - The first 4 entries the logical and physical addresses are equal.
2 - The fifth entry the physical segment is in bank 1.
3 - The sixth and seventh entrys have their physical and logical addresses reversed.
4 - The remaining entries all have physical and logical addresses equal.

GIMIX ENHANCED DAT

In this version of DAT the user can write up to 16 sets of 16 values each to the DAT RAM. Each of these sets of values is functionally equivalent to the 16 values written to the SWTP compatible DAT. Each of the 16 sets represents the DAT configuration for one user or task. Once the required values for all the users (tasks) are written to the DAT a single write to the task select register is all that is required to switch between users (tasks). The 16 locations of the DAT RAM appear at the same memory locations as in the SWTP compatible DAT. Before writing to the DAT, the number ($0-$F) of the particular user (task) to be written must be stored in the task select register. After the task number is written to the task select register, the 16 DAT values for that user (task) can be written to the DAT. To completely initialize the GIMIX enhanced DAT the software must write each of the 16 task numbers to the task select resister, in turn, followed by the 16 DAT values for that task. The task select register is located at $FF7F and like the DAT RAM shares its location with other devices. The task select resister is a write-only device and its logical and physical addresses are always the same.

As in the other version of DAT, the least significant 4 bits of data written to the DAT correspond to the most significant 4 bits of the desired physical addresses, however in the GIMIX Enhanced DAT these bits need NOT BE COMPLEMENTED . They are the true value of the upper 4 bits of the desired physical address. The most significant 4 bits written to the DAT correspond to the bank address of the desired physical segment (bank numbers $O-$F). The least significant 4 bits written to the task select register determine which of the 16 tasks ($0-$F) is active.

Any software that modifies the contents of the task select register must reside in memory that will not be affected by the switching of tasks. It is also important to note that 2 of the remaining

4 bits of this register are used to control other functions on the board. If these functions are in use any programs that modify the contents of the task select register must not inadvertently change these bits. Since the task select register is a write-only device, the last value stored should be maintained in temporary storage for comparison purposes when new data is to be written to the resister. Bit 5 of the task select register is the software control latch for the FPLA address decoder and bit 4 controls the user defined latch output at solder pad "A". See the appropriate sections of the manual for further information on these features.

MEMORY MANAGEMENT CONFIGURATION

The GIMIX 6809+ CPU can be configured for any of the three memory management techniques (STRAIGHT BANK SELECT, SWTP compatible DAT, or the GIMIX enhanced DAT) by the installation of the proper integrated circuits at specific locations on the board. If the board is ordered with one of the two DAT configurations, it is shipped with only the proper parts for that DAT configuration installed. If neither DAT is ordered the board is shipped with only the parts for the STRAIGHT BANK SELECT installed. If the user wishes to change the configuration, of the board, he can obtain the necessary parts and install them at any time.