Using the new EMUL-ST10 Emulator Trace and Trigger Configuration Windows

Application Note # 102 *Version 1.1*

by Robert Boys
Nohau
Campbell, California (888) 886-6428 (408) 866-1820 www.nohau.com

December 6, 1999

Trace Memory and Trigger Mechanism of the EMUL-ST10 (50 MHz)

Introduction

The Nohau EMUL-ST10 is an advanced emulator with speeds of 50 or 80 MHz. This application note describes the trace and trigger configuration menus for the 50 MHz version. The 80 MHz is slightly different and is covered in Application note #103. The EMUL-ST10 supports all ST10 derivatives plus many from the Infineon C166 family such as the C167, C161 and C163. It does not support the Infineon C164 or the C161U.

The EMULST10-PC uses a compact two-board design for decreased weight. One board contains the bondout controller, adapter connections and support logic. The second board is the optional Trace Memory which includes Shadow RAM and Triggers. This note concerns the configuration of the trace board.

The Trace board contains hardware and firmware for the trace memory, triggers, hardware breakpoints, Performance Analysis, Code Coverage, external triggers (in and out), and an general purpose user output connector. The trace board contains its own housekeeping microcontroller. This allows the trace to be started and stopped and other functions to be operated without stealing cycles from the emulation controller. Nohau trace boards can be configured and viewed in real-time and on-the-fly.

Contents

Overall Description

Figure 1 gives an overview of the trace and trigger mechanism. There are four basic trigger mechanisms present in the EMUL-ST10 as shown in Figure 1. They are Level 1 to 6 Breakpoints and Triggers, Level 7 Trigger, Level 7 Filter and the 16 Instruction Pointer Control. This mechanism provides for the hardware breakpoints but not the software breakpoints. The trace card is optional and can easily be added later.

Qualifier data can be entered as addresses, data, SFRs and external signals. They can be entered as numerical hex values or symbols, all with additional values. For example, *timer.sec + 6* is valid. These are combined and compared in each of the four selection mechanisms and sent to the Trigger State Machine. These signals from the selection mechanisms are then combined in an AND/OR Output Array and then sent to the emulator hardware. The External Input signals are sent to each of the six levels of breakpoints if they are enabled.

Levels 1 through 6 and the Instruction Pointers (IP Control) are provided by the ST10 bondout chip. Level 7 Filter and Trigger are Nohau proprietary trace functions. Levels 1 through 6 will be discussed later in more detail. The six levels can be operated independently or can be pre-qualified by the preceding level. *This is important.* This means, for example, that in order for Level 6 to happen, it can be configured such that Level 5 must have already occurred. Or, each level can be independent of the other.

There are two general modes of operation: Standard and Advanced. They are selected by the button found at the bottom right corner of Figure 2. The Standard mode presets some of the Advanced mode settings making setup simpler. The Advanced settings offer all the features of the ST10 bondout. This manual uses the Advanced setting throughout. The Standard setting will remove many of the window boxes from view. These window boxes are preset with default values to simplify trigger setup.

Note the trigger configuration screens were changed during November 1999. The previous mnemonics were changed to intuitive English phrases. A translation table is provided at the end of this chapter. Existing macros will not be affected by these changes.

You can use the triggers to perform these types of operations:

- Break emulation if a specified address and/or data pattern is encountered.
- Record specific CPU cycles in the trace buffer.
- Determine if certain operations take too much or too little time to execute.
- Calculate the time taken for a specified operation such as a function or functions.
- Start and stop the trace recording.
- Send a signal out as certain conditions happen or cause an external input to cause a trigger.
- You can save any number of trace configurations and easily recall them with a button.
- Various combinations of qualifiers are possible to provide sophisticated qualifiers.
- Temporarily “park” a trigger with a simple mouse click.

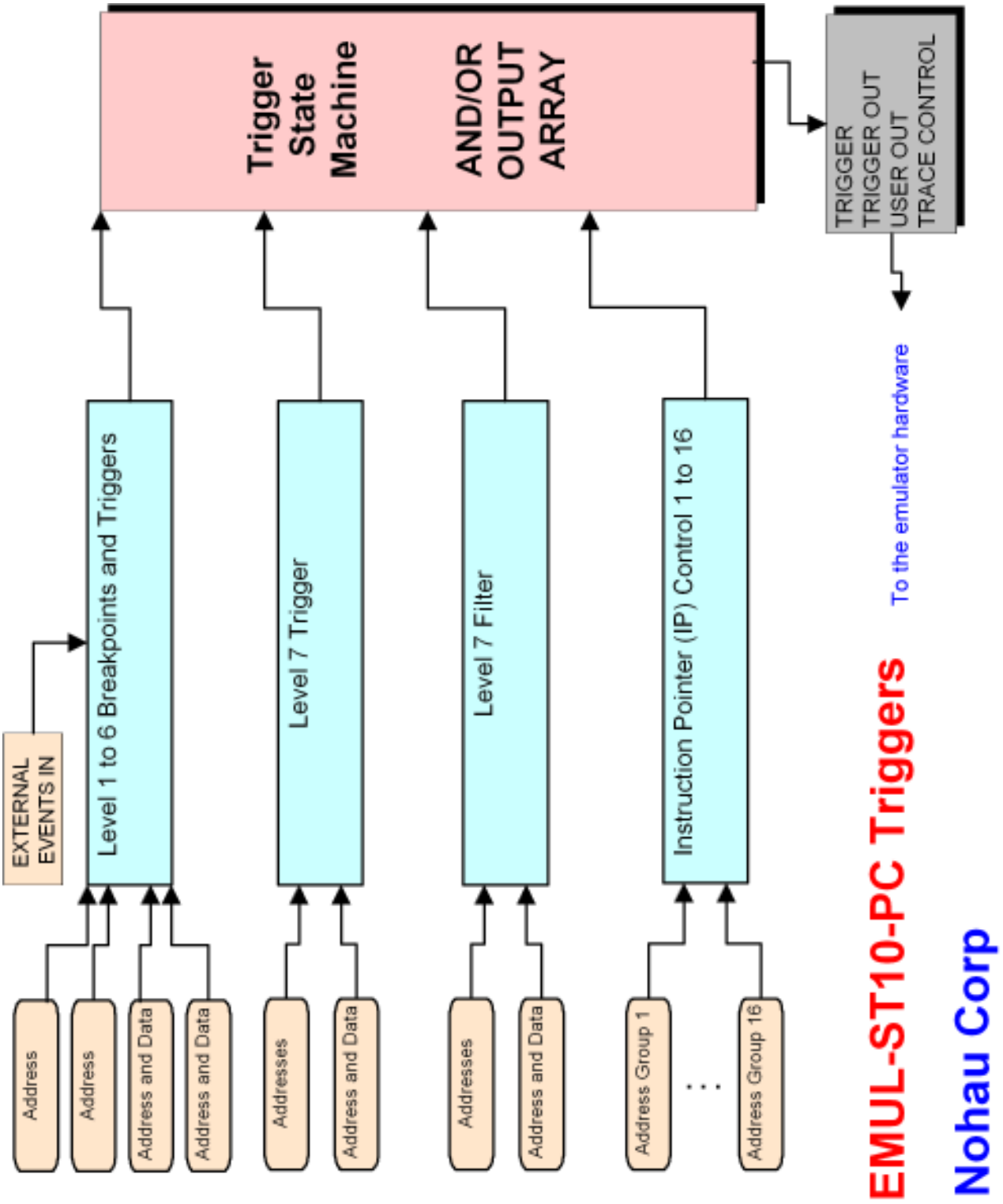
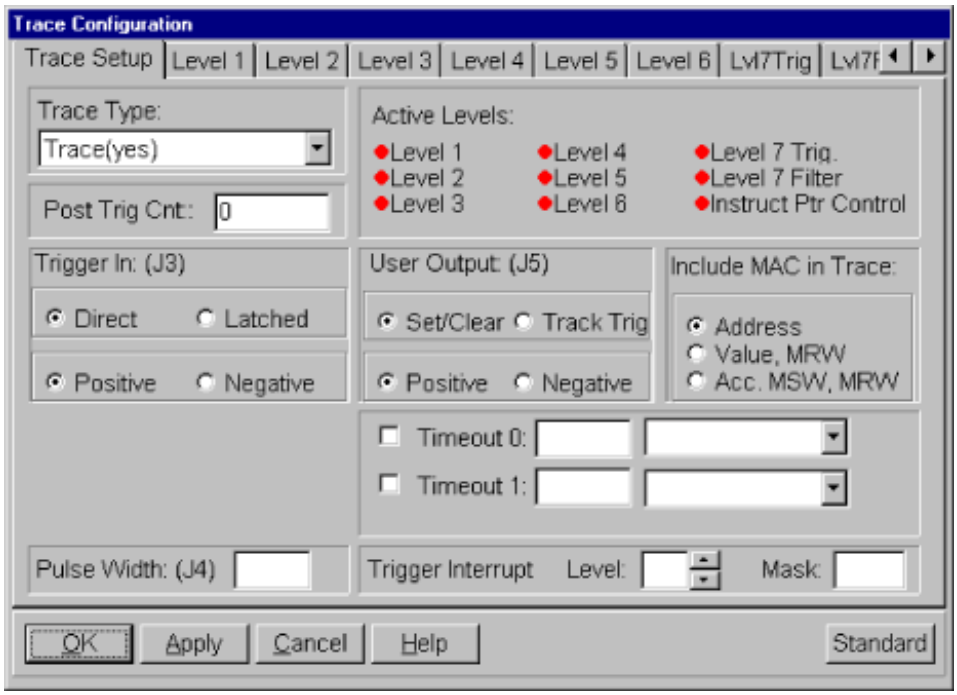

Trigger State Machine
AND/OR OUTPUT ARRAY
TRIGGER
TRIGGER OUT
USER OUT
TRACE CONTROL
To the emulator hardware
EXTERNAL EVENTS IN
Level 1 to 6 Breakpoints and Triggers
Level 7 Trigger
Level 7 Filter
Instruction Pointer (IP) Control 1 to 16
Address
Address
Address and Data
Address and Data
Addresses
Address and Data
Addresses
Address and Data
Address Group 1
Address Group 16
EMUL-ST10-PC Triggers
Nohau Corp

Figure 1

Main Trace Menu

The Trace Configuration window shown in Figure 2 is found under the Config,Trace menu item. It can also be selected by a right mouse click while the cursor is positioned on the Trace window. This window is for the ST Microelectronics ST10 family. Windows for Infineon controllers will be similar. This discussion is for the Advanced mode. The Standard mode has many of these values either not available or have unchangeable default values. Click on the button in the bottom right corner button to switch modes.

Figure 2



Trace Setup

Trace Type:

This check box indicates to Seehau if the trace board is installed. There is one type of trace board so you do not need to specify anything other than it exists or not. In the EMUL-ST10 emulator systems, trace memory is an optional add-on board to the emulator pod. The Trace_TraceBoard command is found in the startup.bas file which is executed when the Seehau software is started and specifies this attribute.

Active Levels:

The status of the various levels is indicated here. A red dot means the specified level is disabled while a green dot indicated it is enabled. A check box in each qualifier in each level tab is used to toggle these values. The various levels are discussed later in this manual.

Post Trig Cnt:

This selects how many frames are recorded in the trace memory after the trigger has occurred. Triggers are programmed to occur when a particular condition is true. Example*: address 500F is written with the value 5E*. This can be viewed as the "when something happens". When this becomes true, the emulator performs an action according to what the user has programmed. This could be to break emulation, start or stop the trace or send an external signal to the outside world. Look at this as "what will then be done".

This can be useful to see what happened not only before the trigger occurred, but also what happened afterwards. If this is set to 7 for example, then the trace will record the cycle of interest plus the next six cycles that follow. The exact number of cycles recorded can vary a small amount depending on the state of the CPU pipeline which is fully decoded in the EMUL-ST10 emulator. A decimal number is entered up to 128K.

Trigger IN: (J3)

An external TTL level signal can be used as an input to cause a trigger. This connector is J3. This input signal can be positive or negative edge sensing according to the setting selected here. The signal can either be latched or free floating (Direct) according to the setting selected. This signal is sampled during each processor frame and cleared at this time.

Trigger OUT (J4)

There is a Trigger output connector (J4) on the trace board that is pulsed when a trigger occurs in a level when the "Trigger - stop trace and pulse Trigger Out" is checked in that level. The trigger OUT will be active even after the trace has stopped.

Pulse Width:

This box sets the pulse width of the pulse sent out J4. This value is 5 bits and is entered 1 to 20 Hexadecimal (1 to 20) and its units are in CPU cycles. A "1" gives a pulse width of approximately 140 ns and 20 gives 2.2 usec. This results from a 50 MHz CPU clock. This sets the USER OUT width also. This effect may change in the future.

User Output (J5)

There is a USER output connector (J5) on the trace board that can be set or cleared according to how it is set in the triggers. This area in Figure 2 sets the characteristics of this TTL level output. This output works even if the trace has been stopped.

Pulse Width:

See Pulse Width above for details.

Set/Clear:

If Set/Clear is set, the value of the USER output will change according to the "Turn User output ON" and the "Turn User output OFF" options in the Level tabs which are discussed later. When this output is turned on for instance, it will stay in this state until the Clear option occurs. Normally two triggers are used to turn the USER off and on. This feature is mutually exclusive to the Track Trig feature.

Track Trig:

Track Trig provides for an output pulse that is active only when the trigger condition is true and the "Turn User output ON" option is selected. The output waveform "tracks" the state of the trigger. This occurs on a cycle by cycle basis. The pulse width is selected in the Pulse Width window.

Positive/Negative:

The polarity of the output can be selected either Positive or Negative and is effective for both the Set/ Clear and Track Trig options.

Timeout:

There are two 12 bit timers used to create a true trigger condition if an event's timing is over or under a user selected number of CPU instruction cycles. Examples: an ISR (interrupt service routine) takes too long (or too short) to occur or takes too long (or too short) in its execution time. These timers are 12 bits in length and values from 0 to 3FF can be entered.

There are two enable click boxes just before the Timeout 0: and Timeout 1: titles. The next box is for the timeout value in instruction cycles of the target CPU. The last box holds values for the prescaler of the timers. The prescaler value is selected by clicking on the arrow and clicking on the appropriate selection. Values are divide by 1, 2^10, 2^20 and 2^30. The timers are selected under the Level tabs.

Trigger Interrupt Level & Mask

This gives the Trigger State Machine a priority within the microcontroller interrupt structure. This setting can be used to block a trigger when a interrupt service routine is at a lower level than the Trigger State Machine. The Trigger State Machine is essentially the entire trigger mechanism of the ST10 bondout controller. The level can be set by priority number (Level) or can be bit selected on a level by level basis (Mask).

Include MAC in Trace:

This selects which information from the ST10 MAC unit is recorded in the trace memory. This box and the selections made are valid only during the execution of a MAC instruction.

Address	The operand address of the MAC instruction is recorded.
Value, MRW	The operand and the MRW (MAC Repeat Word) are recorded.
Acc. MSW, MRW	The Accumulator, MSW (MAC Status Word) and the MRW (MAC Repeat Word) are recorded in the trace buffer.

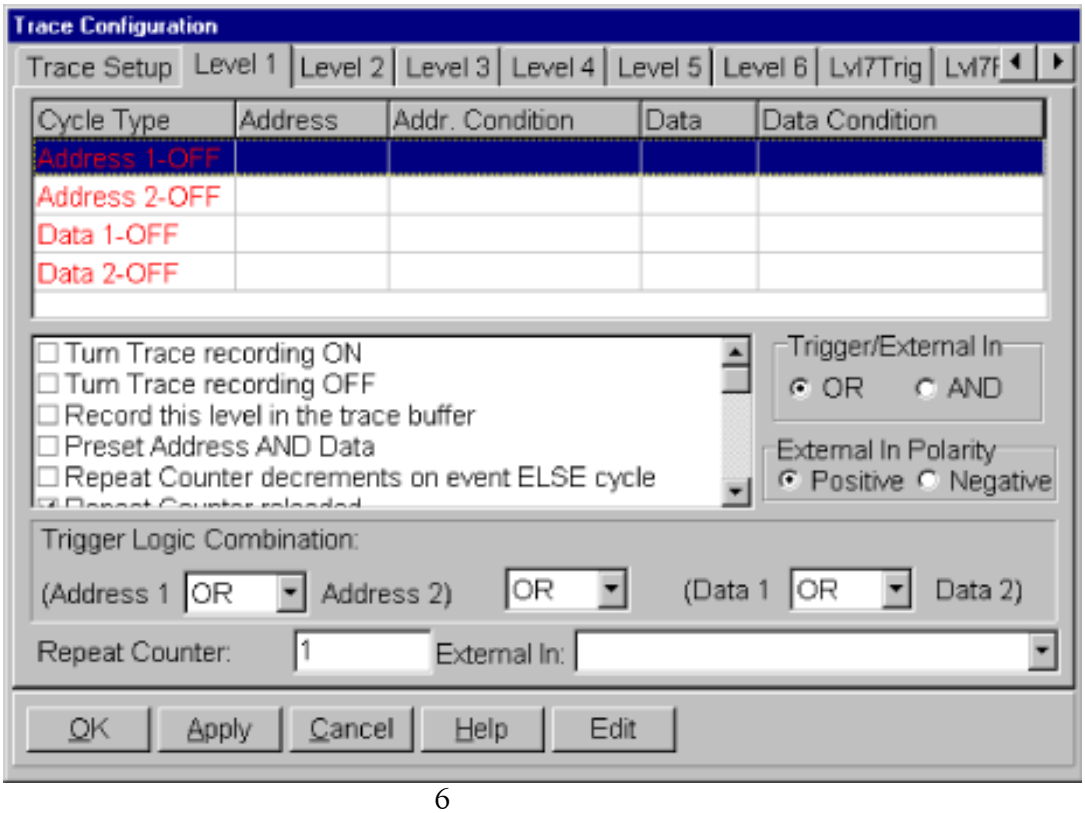
Level 1 through Level 6

Overview:

The ST bondout processor provides for six levels of triggers. These are represented by the six Level tabs in Figure 3. These six levels are provided by the ST bondout chip and are controlled through the Seehau interface. Level 1 is shown selected in Figure 3. Figure 3 is where most of the wording changes have been implemented.

The 6 levels are equal in priority to each other unless the "Previous level must become true first" box is checked. Associated with each level are two data qualifiers and two address qualifiers. These are shown as "Level 1 to 6 Breakpoints and Triggers" in Figure 1. You can include external signals, other levels (even itself), flags, timers and an external input. Figure 4 is a block diagram showing one of the six levels. Figure 5 is the same block diagram but with the "Preset Address AND Data" mode selected. The only difference is the two AND and one OR gates that are selected in Figure 5. This mode allows an address qualifier to be AND'd with a Data qualifier.

Figure 3



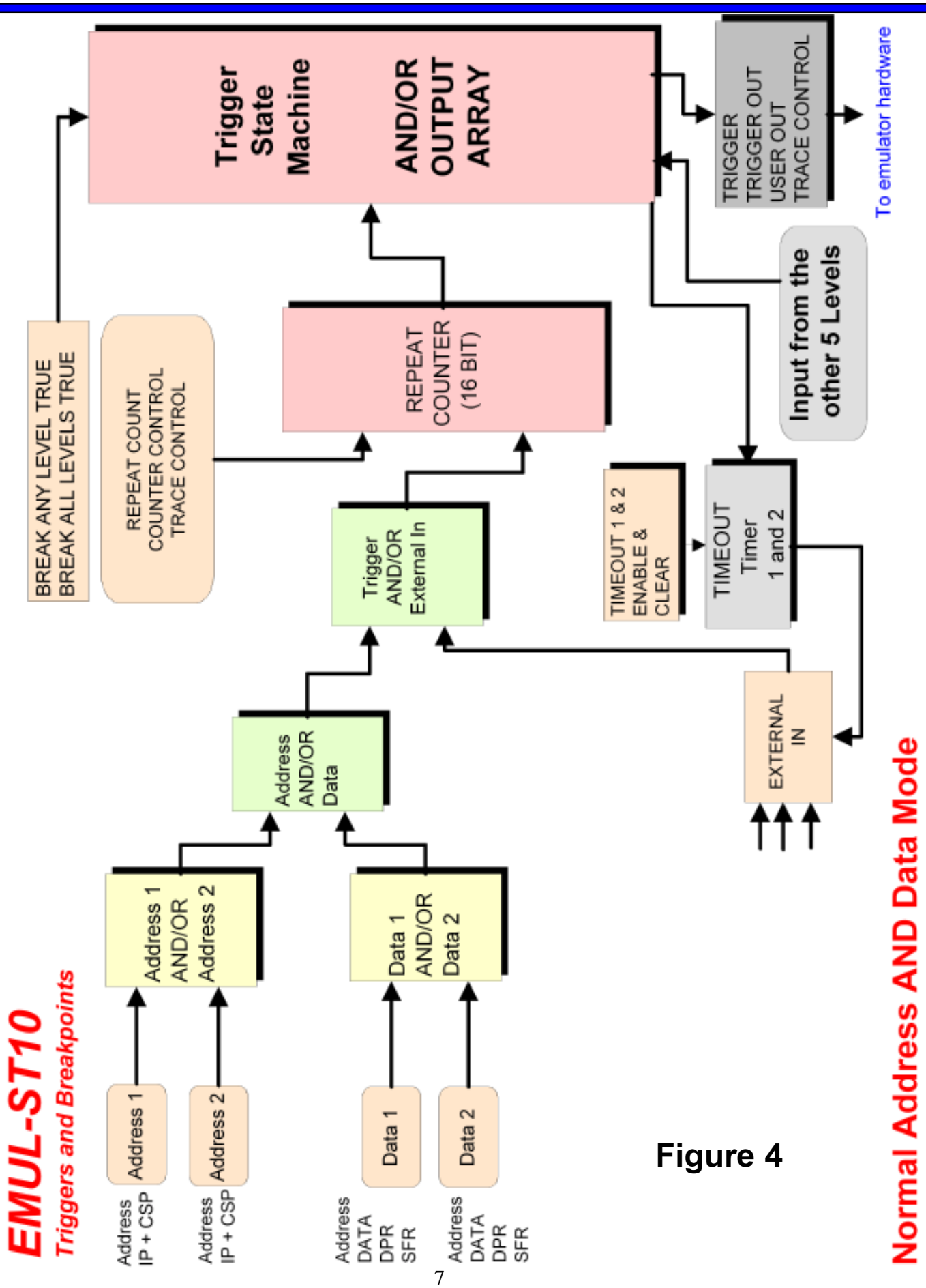

EMUL-ST10
Triggers and Breakpoints
Address IP + CSP
Address 1
Address IP + CSP
Address 2
Address DATA DPR SFR
Data 1
Address DATA DPR SFR
Data 2
Address 1 AND/OR Address 2
Data 1 AND/OR Data 2
Address AND/OR Data
Trigger AND/OR External In
EXTERNAL IN
TIMEOUT 1 & 2 ENABLE & CLEAR
TIMEOUT Timer 1 and 2
REPEAT COUNT COUNTER CONTROL TRACE CONTROL
REPEAT COUNTER (16 BIT)
BREAK ANY LEVEL TRUE
BREAK ALL LEVELS TRUE
Trigger State Machine
AND/OR OUTPUT ARRAY
Input from the other 5 Levels
TRIGGER
TRIGGER OUT
USER OUT
TRACE CONTROL
To emulator hardware
Normal Address AND Data Mode

Figure 4

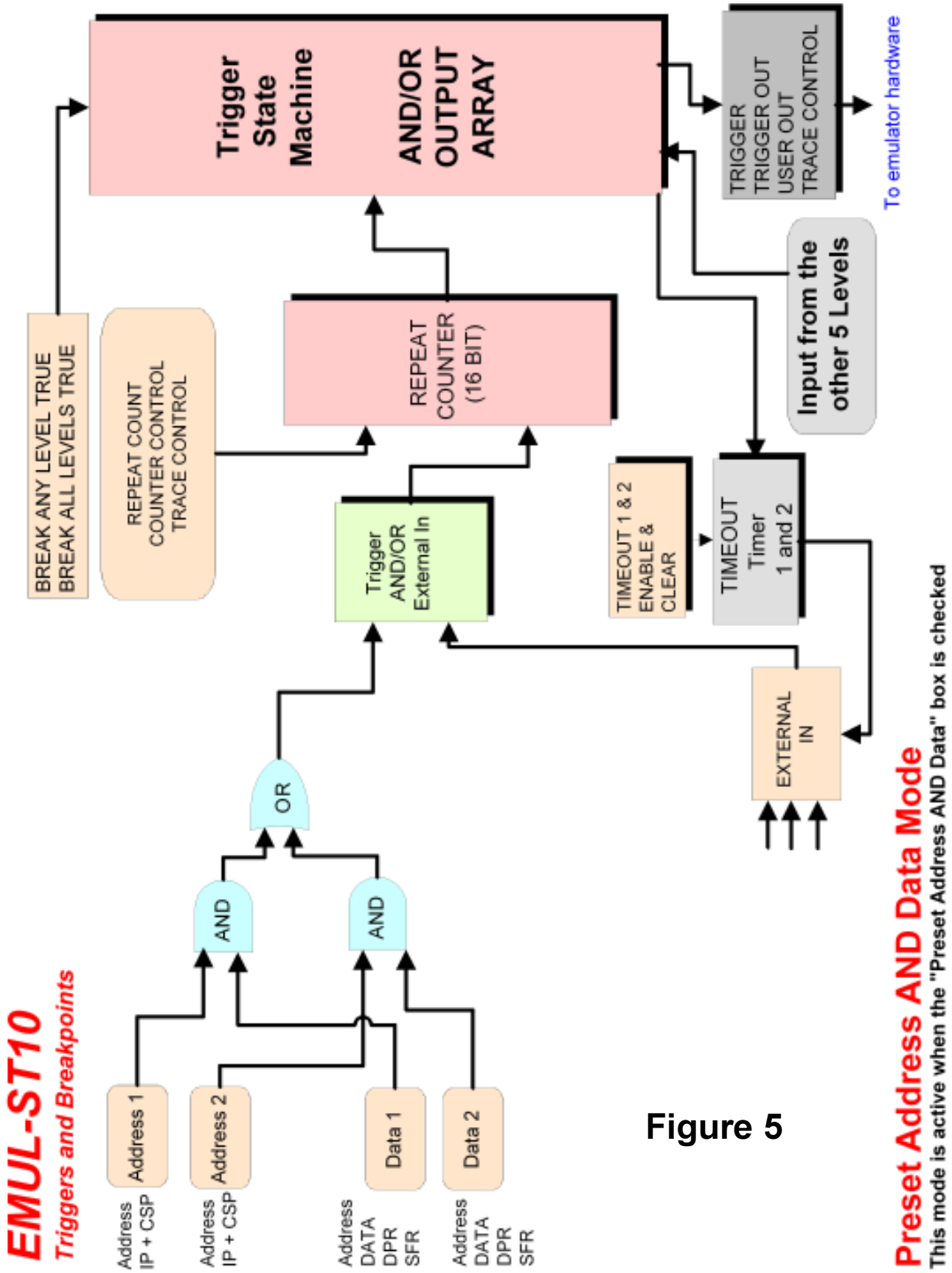

EMUL-ST10
Triggers and Breakpoints
Address IP + CSP
Address IP + CSP
Address DATA DPR SFR
Address DATA DPR SFR
Address 1
Address 2
Data 1
Data 2
AND
AND
OR
Trigger AND/OR External In
REPEAT COUNTER (16 BIT)
REPEAT COUNT COUNTER CONTROL TRACE CONTROL
BREAK ANY LEVEL TRUE
BREAK ALL LEVELS TRUE
Trigger State Machine
AND/OR OUTPUT ARRAY
TRIGGER
TRIGGER OUT
USER OUT
TRACE CONTROL
To emulator hardware
Input from the other 5 Levels
TIMEOUT 1 & 2 ENABLE & CLEAR
TIMEOUT Timer 1 and 2
EXTERNAL IN
Preset Address AND Data Mode
This mode is active when the "Preset Address AND Data" box is checked

Figure 5

Definitions

Qualifier:

The specifications that partially define a condition you want to examine. "If address = 5012" is a qualifier. "If address = 5012 AND data write = FE" is also a qualifier. You can make many types of qualifiers.

Event:

This is the qualifier becoming true. You now choose what is to be done.

Task:

What is to be done when the qualifier becomes an event. When the qualifier becomes true and therefore the event happens: do a specified task. A task is specified by you in the trace configuration window. There are plenty of examples of tasks. To "record this level in the trace" is a good example. A trigger is a signal passed to the trace hardware to tell it to stop recording the trace memory and other to precipitate other tasks.

The trigger is also available as a TTL output signal on a connector (J4) and can be used to trigger an oscilloscope or provide a signal to external hardware. A break is a signal to the emulator hardware to stop the emulation. A task can be a flag used to pass information about an event to another qualifier. An example is when one event has occurred and this fact is needed as input by another qualifier somewhere else. "Previous level must become true first" is a good example.

Note: When most people mention a trigger, they are usually referring to the qualifier or the entire system.

Definitions of the Trace Configuration Menu - Level 1 - 6

Please refer to Figure 3. The names in parenthesis are from the older window and are for reference.

CYCLE TYPE

Address 1-OFF & Address 2-OFF

Each is an address or an address range that are OR'd or AND'd together. Breakpoint occurs before execution for a break event and after execution for a trigger event. These are not the same breakpoints as set in the Source window. Double-click or press the right mouse button to enter or modify data. The OFF shown will turn to a ON when data is entered and the Enabled boxes in the edit windows are checked.

Data 1-OFF & Data 2-OFF

Each is an address or an address range that are OR'd or AND'd together. Events occur after execution for a qualifier becoming true. Double-click or press the right mouse button to enter or modify data. The OFF shown will turn to a ON when data is entered and the Enabled boxes in the edit windows are checked.

TASK OPTIONS WINDOW

These settings activate specific tasks of the trigger system. Most are registers in the ST bondout chip. The bondout mnemonic is indicated in parenthesis and is useful for comparing older documentation and macros. If a box is selected with a check mark, then the associated statement is TRUE.

Turn Trace recording ON (Window (Set))

This setting starts the trace recording when the trigger qualifier becomes true. This is the same as Window mode used in other Nohau emulators. Associated with the trace buffer control.

Turn Trace recording OFF (Window (Clear))

This setting stops the trace recording when the trigger qualifier becomes true. Normally one level is used to turn the trace on and another to turn it off. One trigger is used to start the recording when it becomes true and a second trigger is used to stop the trace when it then becomes true.

This is often used to record the cycles occurring not only inside a specified function, but also those of any functions called by this function. In contrast, the "Record this level in the trace buffer" (Filter Mode) will record only those cycles executed within the function as configured. Combinations with external events such as the Trigger IN are legal. Associated with the trace buffer control.

Record this level in the trace buffer (Filter (Set))

Cycles are recorded in the trace buffer only while the trigger is active. Combinations with external events such as the Trigger IN are legal.

The trace is only recording while a qualifier is true. This mode is address dependent. A range of addresses is set and all cycles executed within this range will be recorded. If a function is specified to be recorded by using a start and end address: calls to functions outside of this address range will not be recorded. When control returns to the calling function within the specified address range, recording will resume. The recording will wrap around once the memory is filled. Associated with the trace buffer control.

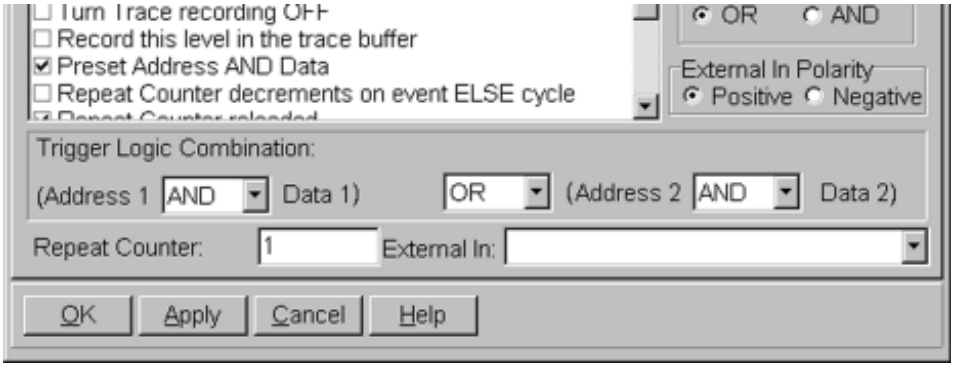
Preset Address AND Data (BC_MOD)

Refer to Figure 4. Note that Address 1 and Address 2 can be AND'd or OR'd with each other. Data 1 and data 2 can also be AND'd or OR'd with each other. These two outputs can then further be AND'd or OR'd together, producing an output that is then fed into the Trigger AND/OR External In logic. These attributes are configured in the Trigger Logic Combination field shown in Figure 3.

Preset Address AND Data provides for a special configuration not provided for in the aforementioned combinations which is shown in Figure 5. If Preset Address AND Data is set, it overrides the settings of the Trigger Logic Combination area. This represents the bondout mnemonics BCMB_IP, BCMB_DA and BC_ID.

Refer to Figure 5. This is the configuration resulting when Preset Address AND Data is enabled. Note Address 1 is AND'd with Data 1. This was not possible with the configuration available in Figure 4. Address 2 and Data 2 are also AND'd together and subsequently OR'd with Address 1 AND Data 1. The Trigger Logic Combination field will not be correct and will have no effect on the trigger configuration logic. Address 1 is swapped with Data 1. A partial Trace Configuration window with the correct information is shown in Figure 6. This issue will be fixed in a future version of Seehau. This is the only entry here that is not a task.

Figure 6



Repeat Counter - a description.

Each level has a 16 bit counter that essentially determines how often an event must happen before the trigger is generated. The value used in this counter is called the Repeat Count and is entered in the Repeat Counter field shown in Figures 3 and 6. For example, If the Repeat Level Count = 4, then the trigger will be activated by the qualifiers becoming true 4 times. If the Repeat Counter = 0, the trigger will not occur. The largest number that can be entered is FFFF. The following three entries are associated with the Repeat Counter.

Repeat Counter decrements on event ELSE cycle (BC_CIC (Event Mode/Cycle Mode))

This selects whether the repeat counter decrements in the Event or Cycle mode. Cycle Mode is a counter decrement on each instruction cycle. Event Mode is a decrement on each event. The event results from a signal which is the input to the counter from the block is labeled "Trigger AND/OR External In" in Figure 5 when a qualifier becomes true. If checked, it is in event mode. Is associated with the Repeat Counter.

Repeat Counter reloaded (BC_CAR (Reload))

This is a binary attribute input defining if the repeat counter should be reloaded with the Repeat Counter value once the counter has cycled and sent a trigger signal to the Trigger State Machine. If this box is checked, the counter will be reloaded once the counter limit is reached and the action can be repeated. If it is not checked, the counter is not reloaded. Essentially, the trigger will then occur only once in this case and can be called a one-shot mode. This box should be checked for general purpose trace and trigger operation. Associated with the Repeat Counter.

Repeat Counter not reloaded while in Event (cycle mode ???) Mode (BC_MCA (Atomic/Cumulative))

This binary attribute input determines if the counter is reloaded or not when qualifier becomes not true when the counter is in the Cycle mode. Cycle Mode is a counter decrement on each instruction cycle and is set by "Repeat Counter decrements on event ELSE cycle" described above. If "Repeat Counter not reloaded while in Event Mode" is checked, it is in Atomic mode. If it is not checked, it is in Cumulative mode.

The input to the counter is the event signal which occurs each time a qualifier becomes true (an event). While this event is true (asserted), each instruction cycle will result in a decrement of the Repeat Counter. If the counter reaches zero, a trigger signal is sent to the Trigger State Machine.

If the counter does not reach zero by the time the event becomes not true, (not asserted), it will be either reloaded (Atomic mode) or not changed (Cumulative mode).

In Atomic mode, the counter must reach zero in the prescribed number of instruction cycles (the counter contents) in order for it to make an event. This is useful if you are looking for events that take longer than this prescribed number of cycles. If the event signal does not stay asserted for long enough, no matter how many times it occurs, an event will not be created.

In Cumulative mode, the counter is not reloaded when the event becomes not true. The next time the event does become true, the counter will continue to decrement form the existing counter value. When the required number of instruction cycles have occurred, even if they are not of any certain duration, an event will eventually be generated. If this box is checked, the counter is reloaded. Associated with the Repeat Counter.

Make this level always true (bypass mode) (FRC_MBE (Force MBE))

This attribute forces the Trigger AND/OR External In signal to be always true, effectively bypassing the selected level. This setting has the effect of making the qualifier true for a given level.

This would be most useful when using the Break Emulation if all levels are true setting. This setting ANDs the Repeat Counter output signals from all six levels in the Trigger State Machine. All six levels must therefore be true for assertion in the outputs of the AND/OR Output Array. This setting forces a specified level to be asserted, effectively forcing it true and removing it from the equation. This happens if the box is checked.

Previous level must become true first (PL_CMB (Previous Level Dependence))

This is the only time a level has a priority over another level. This allows one level to affect the next level. In order to have a level affect any other level (i.e. not the next level), select the appropriate action in the External In: box shown in Figure 3. In order for a level to become true, the previous level must also be true. For example, if Level 5 has this attribute set, Level 4 must be true in order for Level 5 to become true. Level 1 can never be conditioned with this attribute since it is the first level. Check this box to enable this feature.

Trigger State Machine - a description.

This mechanism feeds signals from each level to the AND/OR array which determines how the Repeat Counter signals will be used. There is only one AND/OR Array in the emulator and all levels share it. Each Repeat Counter output for any individual level is used for the Trace Control, USER OUT and the Trigger OUT connectors. The Trigger signal (not the same signal as Trigger OUT) is used as a signal for the rest of the emulator hardware and is generally used to stop the trace recording. The following twelve attributes found in the Task Options window configure the Trigger State Machine. They are actually registers in this machine and there is a set for each of the six Trigger State Machines. See Figure 4 for the block diagram.

Trigger - stop trace and pulse Trigger Out (Trigger (Set))

This attribute sets the trigger to be active when the associated qualifiers become true. This trigger signal is used to activate the specified task and is provided on the trigger output connector (J4) on the ST10 trace board. This attribute is always in Track Trig mode therefore the output signal tracks the event. Check the box to activate this feature.

Turn User Output ON (USER (Set))

This attribute sets the user output connector (J6) on the ST10 trace board. This output will go high when a qualifier is asserted. This output will be latched high until turned off. Check the box to activate this feature.

Turn User Output OFF (USER (Clear))

This attribute clears the user output connector (J6) on the ST10 trace board. This output will go low when a qualifier is asserted. Check the box to activate this feature.

TIMEOUT - a description

There are two 12 bit timeout counters 0 and 1 in the AND/OR array. This means they are shared by all levels. They are set in the Trace Setup window. See Figure 2. The timeout commands are always in the Set/ Clear mode. These timers clear the Repeat Counters when they count up to 3FF and if enabled for a given level. The timeout counters are connected to the particular level via the External In dialog box found in Figure 3. The Timeout timers get their clocking input from the output at the Trigger AND/OR External In logic which becomes true when a qualifier becomes true. The timer load value and prescaler is entered in Figure 2.

Enable Timeout 0 (TIMEOUT 0 (Enable))

This attribute is used to enable the timeout counter 0. It is loaded with the value set in Figure 2. The counter increments each time an event happens. It clears the Timer Counter when it reaches 3FF if "Timeout x clears the Repeat Counter" is enabled.

Disable Timeout 0 TIMEOUT 0 (Clear))

This attribute is used to clear the timeout counter 0 when a specified event happens.

Enable Timeout 1 TIMEOUT 1 (Enable))

This attribute is used to enable the timeout counter 1. It is loaded with the value set in Figure 2. The counter increments each time an event happens. It clears the Timer Counter when it reaches 3FF if "Timeout x clears the Repeat Counter" is enabled.

Disable Timeout 1 TIMEOUT 1 (Clear))

This attribute is used to clear the timeout counter 0 when a specified event happens.

Timeout 0 clears the Repeat Counter (LC_CT0)

This enables Timeout 0 to clear the Timer Counter when it reaches 3FF.

Timeout 1 clears the Repeat Counter (LC_CT1)

This enables Timeout 0 to clear the Timer Counter when it reaches 3FF.

Break emulation if this level is true (Break (Any Level True) (OR))

When the event happens (i.e. the qualifier becomes true) emulation will be halted and the trace board will be stopped. This attribute is OR'd with the other level break emulation attributes.

Break emulation if all levels are true (Break (All Levels True) (AND))

When the event happens (i.e. the qualifier becomes true) emulation will be halted and the trace board will be stopped IF all the other levels are true. This attribute is AND'd with the other level break emulation attributes. All the six levels must be true for emulation and trace recording will be stopped. Recall a specified level can be turned off with the "Make this level always true (bypass mode)" attribute.

Entering Data into Level 1-6

Figure 7 shows the Level 1 Trace Configuration window. This is similar to Figure 3 but has a qualifier entered in Data 1. Each level has 2 address and 2 address/data qualifiers and they are shown here. The ON or OFF in the label field indicates if this particular qualifier is enabled or not. Figure 7 shows a Data qualifier enable for a write of 3400 to address 5017.

To enter or edit data either double click on required address or data line or right mouse click on the line. The right mouse click also enables you to remove the line. Figure 8 will open up. Figure 8 is for Data: Address has the data windows greyed out and a new checkbox called Force Sync Delay. Force Sync Delay is used only for the address qualifiers. It is likely that the address event (i.e. an address match) will occur at a slightly different time therefore the combined event will not occur. The first event is delayed to coincide with the second event. This effect is automatically enabled for the two Data qualifiers.

If the two enable check boxes are unchecked, the qualifier will be disabled, the word OFF will appear in Figure 7 and if the other 3 qualifiers are disabled, the green enable light will go off in Figure 2.

You can type the symbol name in place of the 5017 and show + 7 can also be used. You can also copy the symbol name from the symbol browser found under the View menu item. Note the various combinations that can be entered in this window. Examples using these windows are given at the end of this chapter.

Figure 7

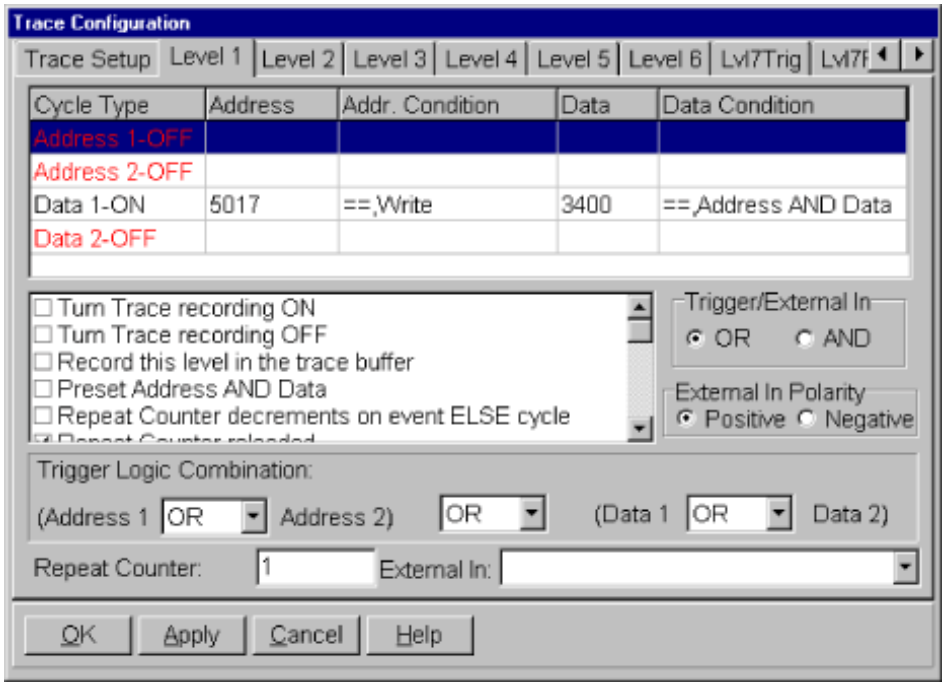
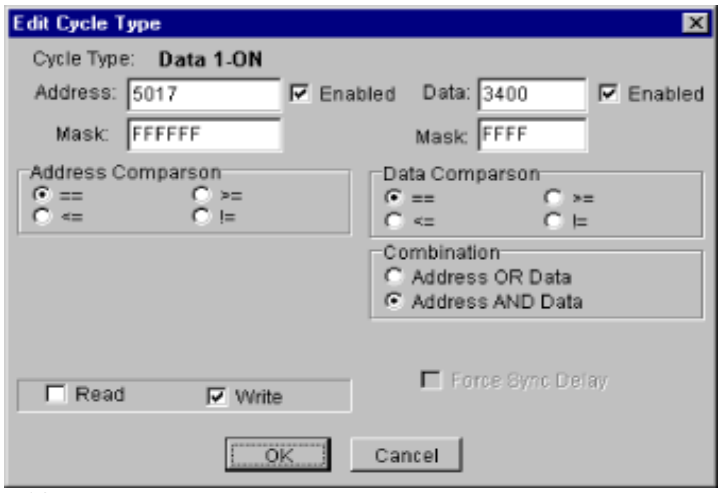


Figure 8



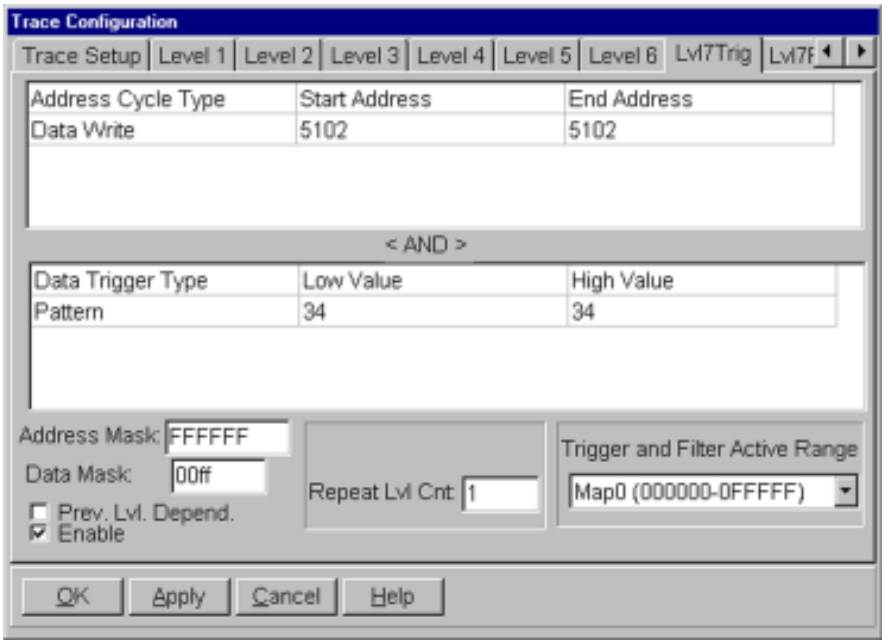
Level 7 Trigger

The Level 7 mechanism was developed by Nohau to provide additional trigger abilities to the ST10 bond-out. Level 7 operates on data read and write cycles and not on instruction fetches. This type of data access is easy to implement in the trigger mechanism and use because it does not use the instruction queue as an address access would. The number of address and data combinations that may be entered is virtually unlimited but is purposely limited to around fifty for practical reasons. Level 7 will only stop the trace recording and not break.

Level 7 Trigger is used to cause a break when the specified qualifiers become true. Level 7 Filter is used to allow only those R/W cycles specified to be recorded in the trace memory and is discussed later.

Figure 9 shows the Level 7 Trigger tab window selected from the Config, Trace menu item.

Figure 9



Address Cycle Type

The read or write address qualifier is specified in this field. The address field can contain a single address (as shown) or a range of addresses. Addresses are entered, edited or removed by highlighting the appropriate line and doing a right click on it or double-clicking on an existing line which opens the window in Figure 10. Note the addresses have already been entered. Note the various data accesses that are allowed.

Data Trigger Type

The data qualifier is specified in this field. Data is entered, edited or removed by highlighting the appropriate line and doing a right click on it. The window in Figure 11 will open up. This field can contain a range or a pattern of data. Recent values are available under the down arrow. Pattern specifies only one value to be placed in the Begin field. Range uses both fields to specify a value range. If this field is empty, then the data value will be don't care when the address specified in Figure 10 becomes true.

Figure 10

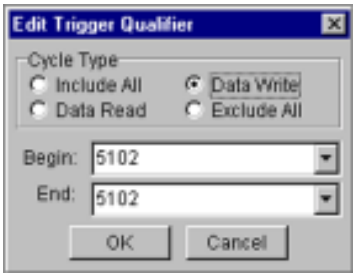
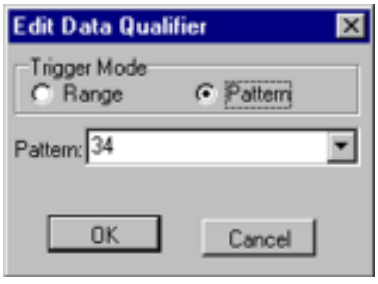


Figure 11



Address Mask

All address comparisons are done on the full 24 bit address. This field allows masking of the address on a bit by bit basis and is entered as a hexadecimal value. A bit set to zero will give that position a "don't care" value. A one will make that position relevant to the address comparison. This mask is global to all addresses listed.

Data Mask

This allows bit positions to be given a "don't care" status similar to the Address Mask.

Prev. Lvl. Depend.

Previous Level Dependency allows Level 7 to be connected to Level 6. Level 7 will generate an event only if a Level 6 is also true.

Enable

Enables or disables the Level 7 Trigger. Ideal for temporary parking of the triggers without deleting them. This box must be checked to allow the filters to be active.

Repeat Lvl. Cnt:

The Repeat Level Count is a 16 bit counter that determines how often an event must happen before the trigger is generated. The value used in this counter is called the Repeat Level Count. If the Repeat Level Count = 4, then the trigger will be activated by the qualifiers becoming true 4 times. This is not the same counter as the Repeat Counter in Levels 1 through 6 but it performs the same function.

Trigger and Filter Active Range

This field is provided to prevent aliasing or memory wraparound issues from affecting the triggers. The triggers will be effective only if the qualifiers become true in the specified range.

Notes on Address and Cycle Type & Data Trigger Type Combinations

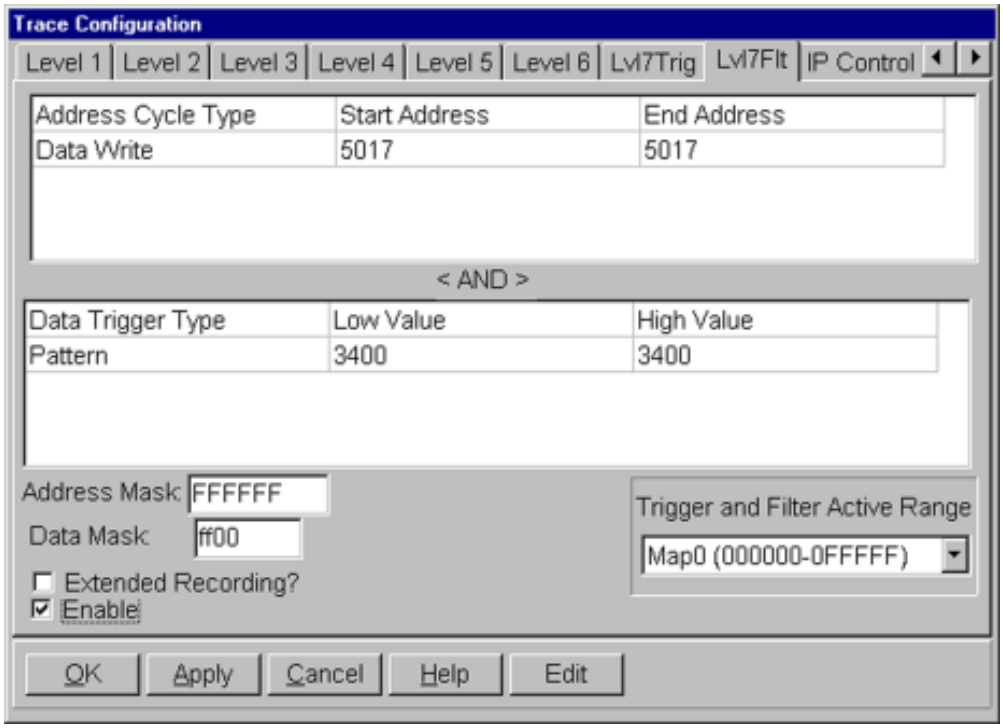
There can be 50 addresses and 50 data values. The addresses are all OR'd together. The Data values are all OR'd together with themselves. These two groups are then AND'd together. This means that for setups with more than one entry in each type, a given address is not paired up with a given data value. If there is only one entry in each of the two windows, they are by default paired with each other. An example using Level 7 Trigger is given at the end of this chapter.

Level 7 Filter

Level 7 Filter is used to allow only those R/W cycles specified to be recorded in the trace memory. The Level 7 Filter is very similar to the Level 7 Trigger in its configuration. Review the section on the Level 7 Trigger. This Filter is functionally equivalent to the "Record this level in the trace buffer" task.

The window shown in Figure 12 in under the Lvl7Flt tab in the Trace Configuration window. Qualifiers are entered the same way as for the Level 7 Triggers. All local menus and fields are identical.

Figure 12



Extended Recording?

This field allows for the trace memory to continue recording for a specified number of cycles after the trigger has happened. This captures some cycles after the trigger event has occurred.

The filter settings shown in Figure 12 result in the trace display in Figure 13. Note the writes of 3400 to address 5017 by the MOVB instruction with an opcode of B92C located at address 442.

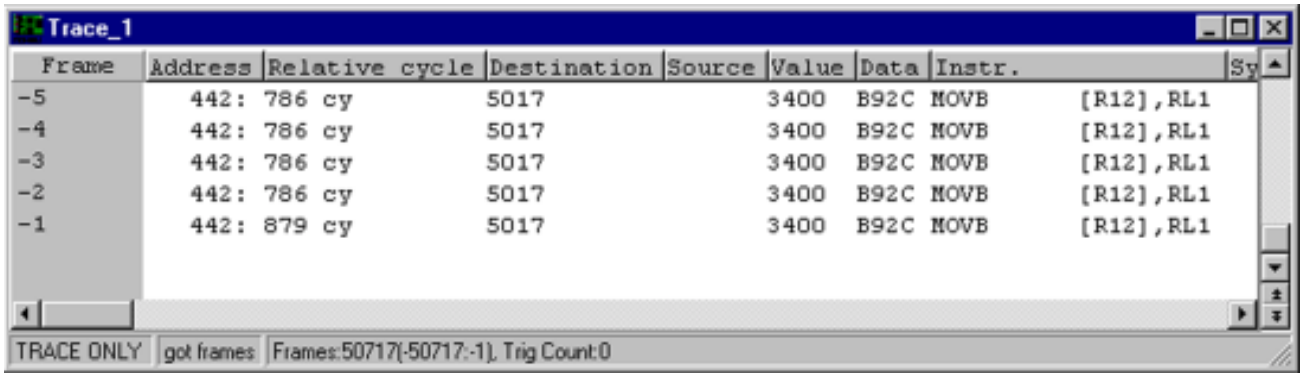


Figure 13

IP Control: Instruction Pointer Control

The last tab in the Trace Configuration menu is the IP Control breakpoints. This is the bondout facility used for the hardware breakpoints used in the Source window. There are 16 groups and each group consists of 2 address qualifiers. These two address qualifiers can be used for two address comparisons or one range comparison with the CPU Instruction Pointer. Data values are not used as qualifiers. Groups are only available in the Advanced Trace mode.

The IP Control tab in the Trace Configuration window is shown in Figure 14. This window is available under the Config, Trace menu item. An example address is entered in Group 1.

Group

These 16 fields contain addresses used as qualifiers. Each group is OR'd together.

Group Condition

This qualifies the 16 Groups to Levels 1 through 6 events. This is global to all 16 group members. The specified Level must be true for any Group member to become true.

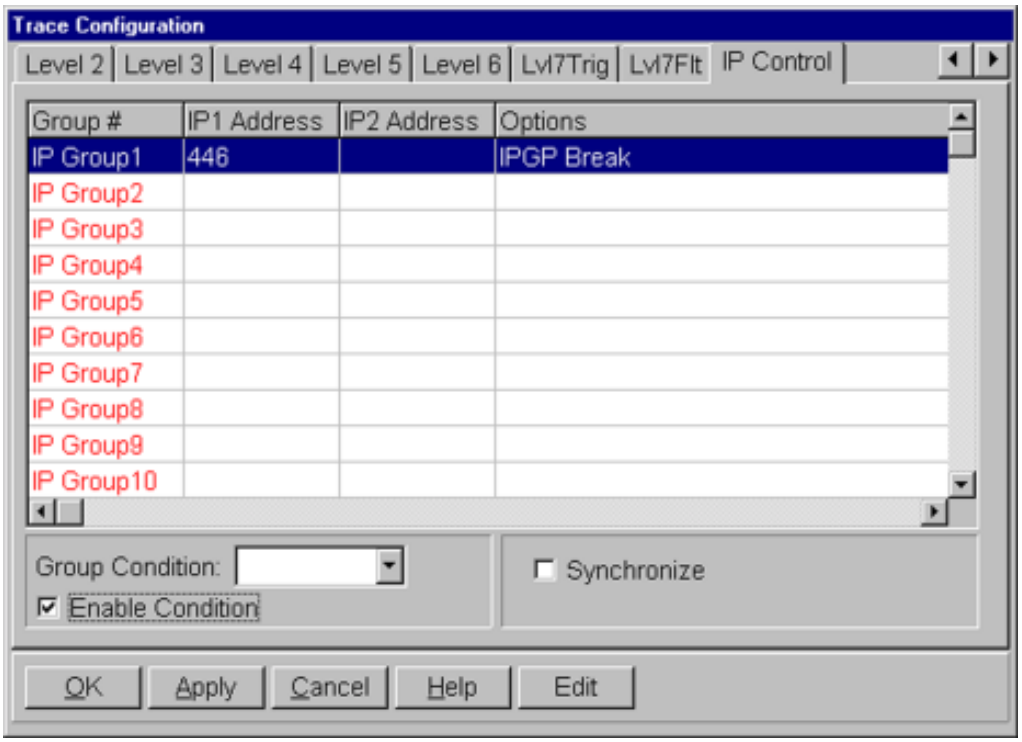
Enable Condition

This enables or disables the Group Condition.

Synchronize

This enables or disables the synchronization of address events with data qualifiers from another level if so selected. This ensures that the address creating a data action will correctly be AND'd with that data qualifier. This is in spite of each qualifier potentially occurring at slightly different times.

Figure 14



Editing a Group Member

Click on one of the Group Type to highlight its line. Double-click on this line and Figure 15 opens up. This window is where the address qualifiers are entered. You can also right click on it to get the remove and edit functions. A simple example is provided at the end of this chapter.

IPG0 Address: and IPG1 Address:

IPG0 and IPG1 are the two address qualifiers. Each can selectively be enabled with the Enable check box.

Range:

The Range mode compares IPG0 and IPG1 with the current CPU instruction pointer following this equation:

IPG0 =< current IP < IPG1

If a match occurs, the following actions can be selected:

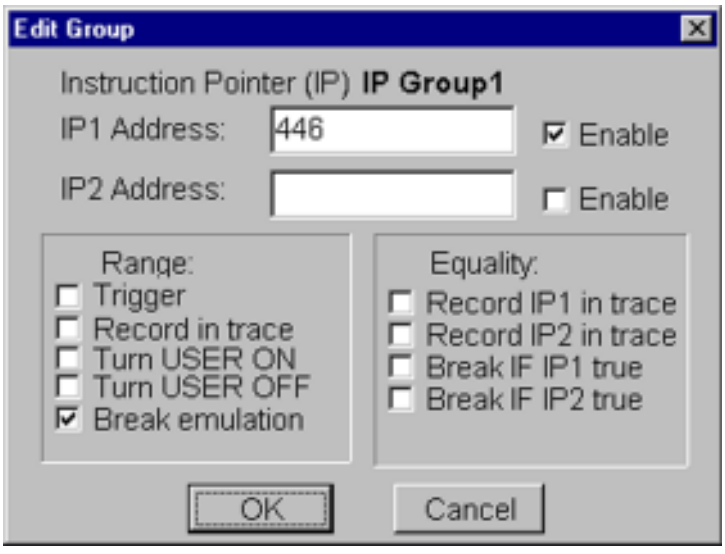
Trigger:	This is a signal that causes the trace recording to stop.
Record in trace:	This signal is used to turn the trace recording on while the qualifier is true.
Turn User ON:	This signal is sent to the USER Output connector J6 on the Trace board. USER is set high by this selection. See Figure 2.
Turn User OFF:	This signal clears the USER Output connector J6.
Break emulation:	This selection causes the emulation to stop.

Equality:

IPGP0 and IPGP1 can be configured as the Range or Equality mode. Equality is comparing IPGP0 and IPGP1 with the instruction pointer and creating an event if a match is made. An event will create an action. These actions are listed and are shown in Figure 15 under the Equality: title.

Record IP1 in trace:	When IPGP0 becomes true, the cycles responsible are recorded in the trace memory.
Record IP2 in trace:	When IPGP1 becomes true, the cycles responsible are recorded in the trace memory.
Break if IP1 true:	When IPGP0 becomes true, the emulation will be stopped.
Break if IP2 true:	When IPGP1 becomes true, the emulation will be stopped.

Figure 15



Saving the Trace Configuration

The entries you have made can be saved to a macro. This macro can be started manually under the Macro, Run menu item or a button can be created on the toolbar. There are two methods of making a macro:

1) Manually by creating the text file filename.bas

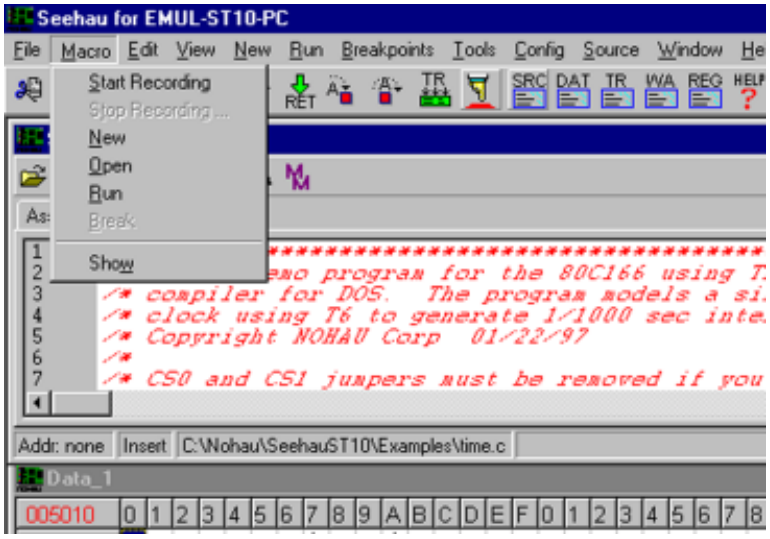
This method is not recommended as a detailed knowledge of the macro commands will be needed. It is easier to make this *.bas file with the macro recorder found in the Macro/Start Recording menu. Manually editing an existing file is easy and recommended with the built-in macro editor under the Macro, Show menu item or with any ASCII text editor.

2) Using the built-in Macro Recorder

In the Macro menu item, there are two items used to start and stop the macro recording. All your key-strokes will be saved to a filename.bas file of your choosing. Such a file is run either with a button you create or manually with the Macro, Run menu item. See Figure 16.

A macro can also be created from a configuration window that has an Apply button. The present settings of the configuration menu will be saved at one time. This is useful for saving a series of settings that have been subsequently modified and the exact key sequence is not available.

Figure 16



Quick Saving a Configuration Macro using the Apply Button

1) The configuration menu must be open and the Apply button visible. Figure 12 is an excellent example.
2) In the Macro menu item, select Start Recording. The configuration window will disappear.
3) Reopen the configuration window from the appropriate menu item. Click on Apply. The settings associated with this window will be saved.
4) In the Macro menu, select Stop Recording.
5) The Save Macro window will open giving you the opportunity to choose the filename for the newly created macro. Enter a filename of your choosing and click on Save. The macro is ready to use and will accurately recreate your configuration settings. You can also easily make a button to activate your macro. This feature is found under Config, Buttons.

Trace Memory Examples

1) The timer example as described in the Getting Started manual and repeated in Figure 17 showed the Shadow RAM providing memory contents in real-time. Recall there is no cycle stealing from the emulation controller. While the time.c program has been running, the Trace Memory has been operating in the background, also in real-time.
2) Many emulators are unable to view the trace without stealing cycles or even stopping the emulation. The Nohau emulator can do this in real-time. It uses a dedicated microcontroller to do all the Trace, Trigger and Shadow RAM housekeeping chores, rather than stealing cycles from the bondout controller.
3) Ensure the emulator is running the time.c program. The file you want to load is time.abs. The two boxes in the bottom left hand corner of the main window will contain "Running"instead of "Stopped" as shown in Figure 17. The GO and TRACE icons are both red in color. You should have the Data window open and displaying the time changing in real-time as shown in Figure 17.

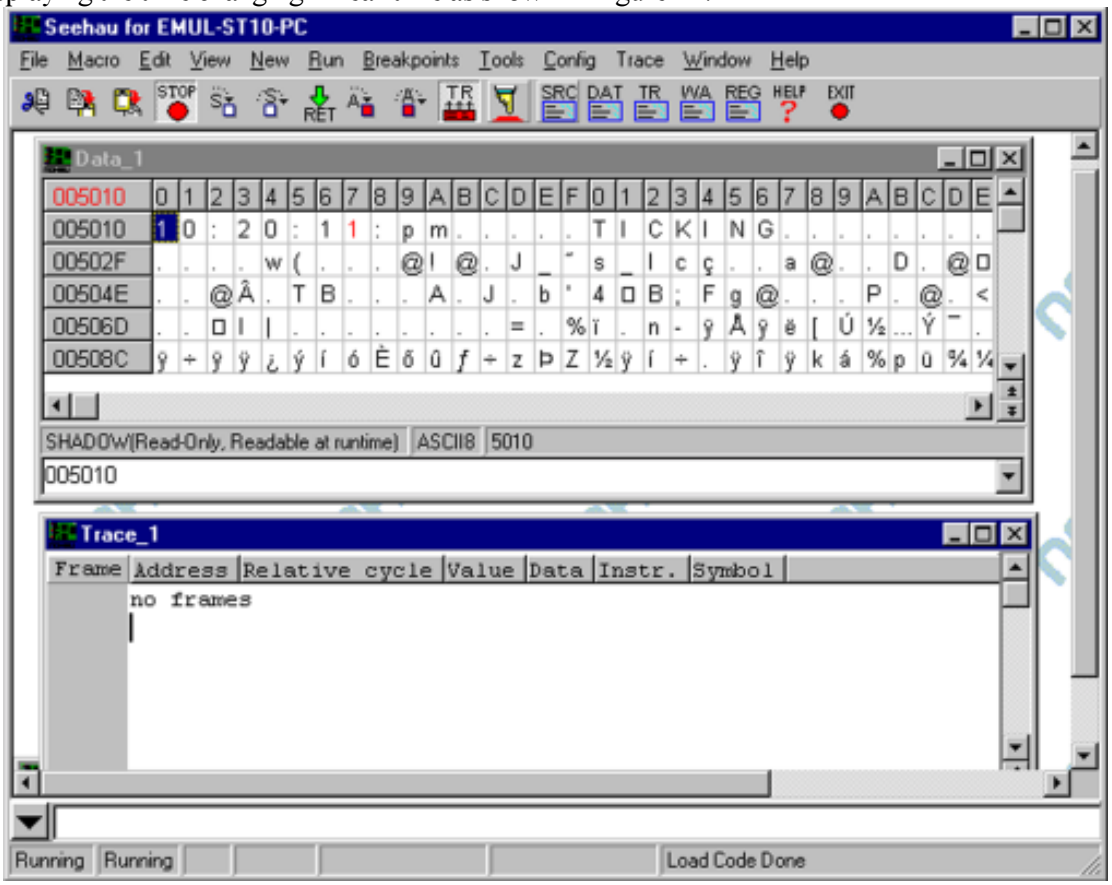


Figure 17

4) Open up the Trace window if necessary. You can click on the New Trace icon or in the menu New, Trace.
5) Position the windows similar to Figure 13. You should have the Trace and Data windows visible. The trace window might have some data recorded in it or empty. This depends on previous emulation runs.
6) Note the Trace window is empty as the Trace buffer is being filled. It is not possible to view the Trace contents at this time. The dialog bar at the bottom of the trace window will show that the trace memory is already full and how many triggers have occurred. There are no trigger events. See Figure 14.
7) Click on the Stop Trace icon. Note that while the changing time values may appear to stumble: this is only in the display. The emulation controller was not slowed down at all. This is genuine real-time operation.
8) The Trace window now contains recorded controller cycles. Figure 18 shows the Trace Memory that I got. I added some columns by right clicking on the Trace window and selecting them. Add your own.

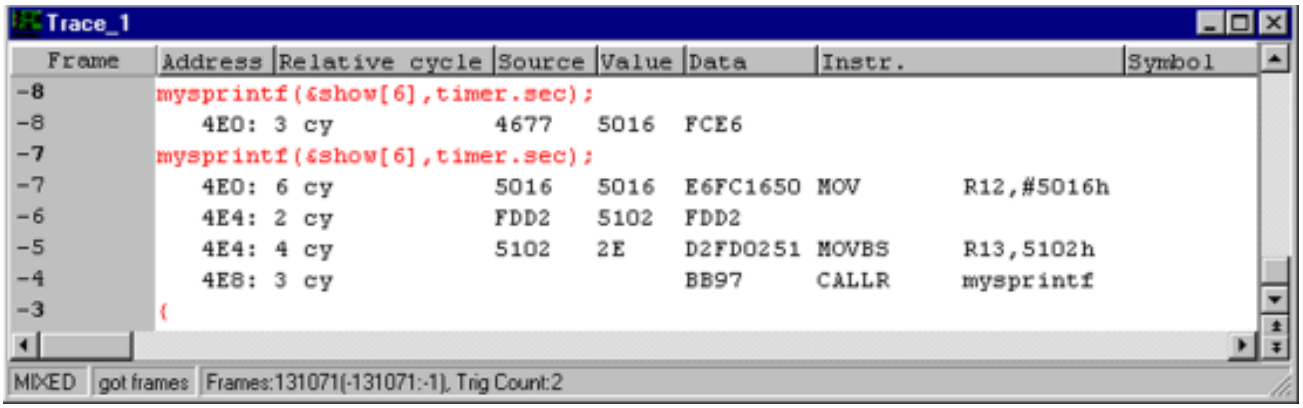


Figure 18

9) Note that in Figure 18 labels, registers, source code and addressing modes are all displayed. Note the trace window will have the ability to display C source code with the resulting assembly code.
10) Start the Trace memory by clicking on the Start Trace icon. Note the time does not stop or slow down. Once again, the trace memory is being continuously overwritten with new values. The trace memory is a circular buffer. This will continue until the recording is stopped either manually or with a trigger event. Note the triggers have the ability to start and stop the trace recording.

The trace window can display many types of information about the bus cycles. Right click on the trace window or open up the menu in Config, Trace. The window in Figure 19 will appear. These are the options used to create Figure 18. Click on some of the options to see the various fields available. The source code can be enabled under Display mode.

The ST10 emulator has extensive trigger facilities to start and stop the trace as well as perform program breaks. The triggers can control cycle recording so that only specified cycles are recorded. The apparent size of the trace memory can be magnified many times by using carefully selected trigger qualifiers. The triggers can also send out external signals.

External events present on J7 (see the trace board layout in the hardware manual) can be recorded in the trace (8 bits).

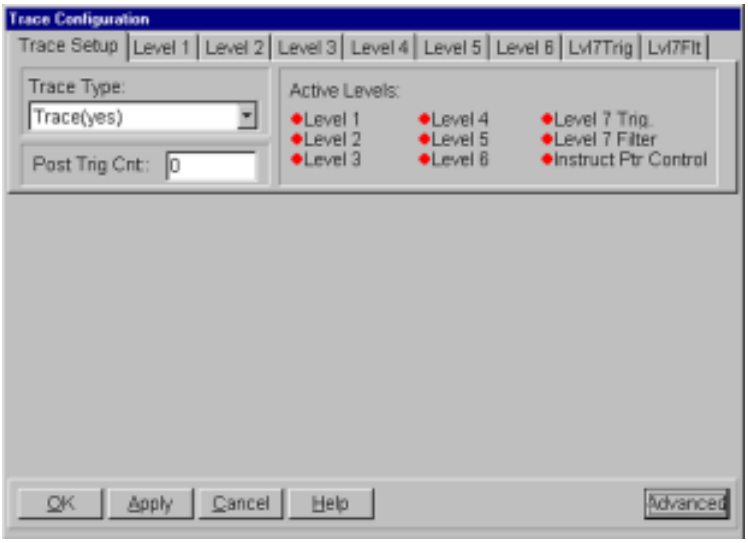
Figure 19

Setting a Level 1 - 6 Example Trigger

All Nohau emulators contain sophisticated and versatile triggers. The EMUL-ST10 offers unparalleled trigger capabilities. You can configure the triggers in real-time. The triggers test for qualified events, perform tasks and start and stop the trace all without stealing cycles from the emulation controller. The application program always run full speed. A break in emulation, by definition, affects real-time operation. This example will show how to setup a simple yet effective trigger. The emulation will stop when the seconds MSB memory location is written the value of 4 ASCII (34 Hex). The bus cycle responsible for this write will be recorded in the trace memory.

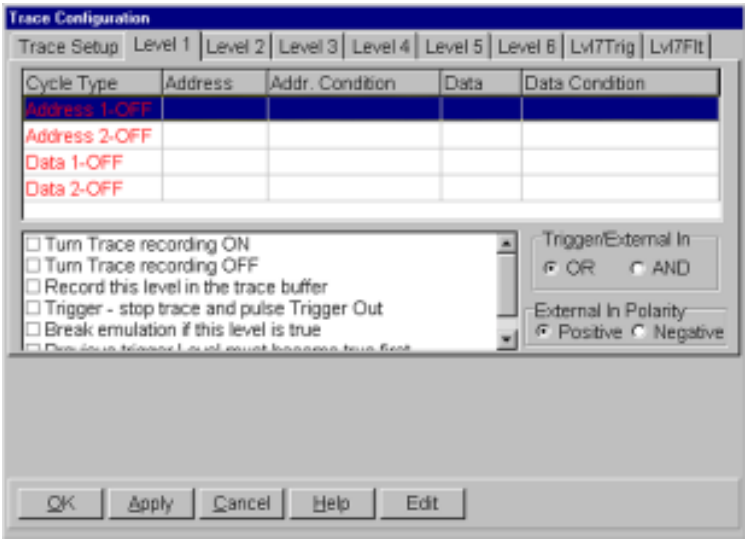
1) Configure Seehau to give a main window similar to Figure 17. Stop emulation and press the Reset icon.
2) Open the Trace Configuration menu under Config, Trace or right clicking on the trace window and selecting the Trace Config item. The window in Figure 20 opens up. Note the button labeled Advanced. There are two versions of the trace window for the ST10 emulator: Standard and Advanced. Click on this button to see some of the Advance features. We will use the Standard mode here, so click again so it the box reads Advanced and the Standard mode is therefore selected.
3) Note there are 7 levels of triggers plus a Group (Advanced only). We will use only Level 1 in this example. The red dot will change to green once we have entered and enabled the qualifier for Level 1.

Figure 20



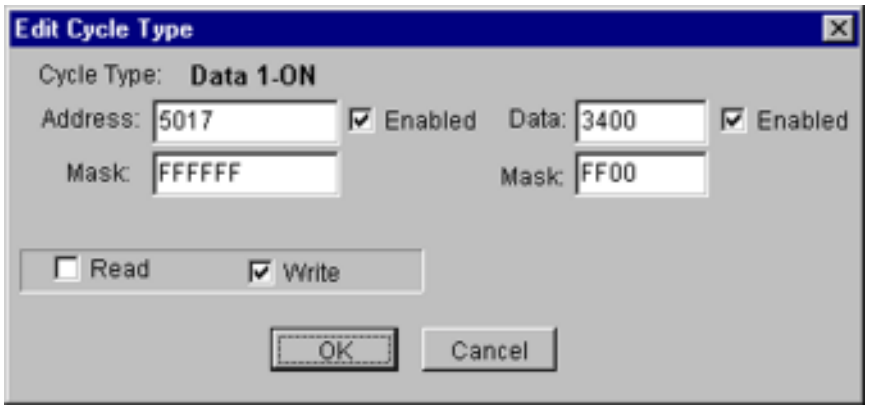
4) Click on the Level 1 tab and the window opens up in Figure 21. Address 1 and Address 2 are address only qualifiers. Data 1 and Data 2 are address and data qualifiers. We will use Data 1 and Data 2 in this example. All qualifiers are disabled at this point.

Figure 21



5) Click on the Data 1 line and right click (or double-click) on the highlighted line. An empty Figure 22 will open up. Enter the fields as shown and ensure the Enabled boxes are checked. Address and data are not case sensitive. You need to fill in the Address, Data, Data Mask and the Write boxes. Press OK. Data 2 will automatically be filled in with the same data as Data 1.

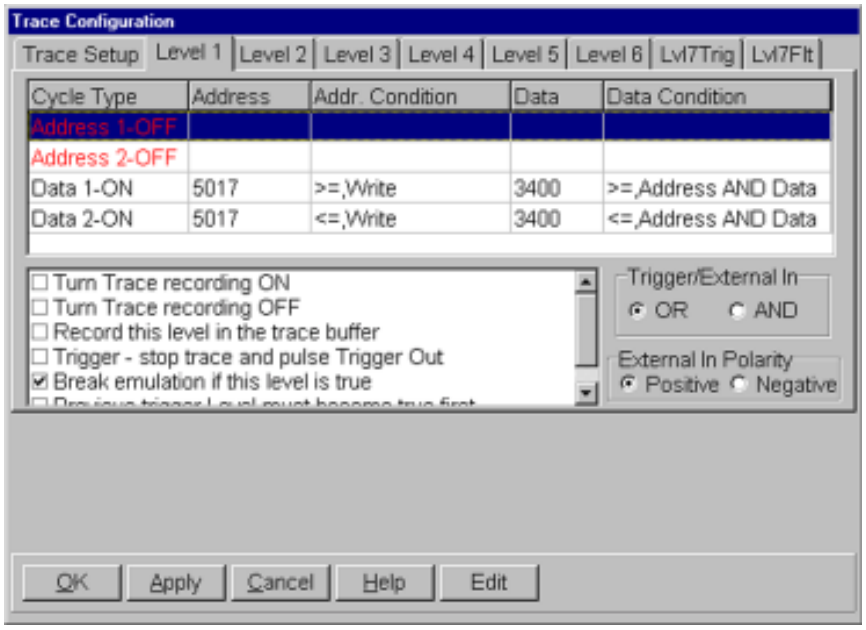
Figure 22



6) Figure 23 will appear. Note that both Data 1 and Data 2 are enabled with the appropriate data and are turned ON. The qualifier is now properly set. Data 1 has a window similar to Figure 18. Click on the Data 2 line and right click as before to open this menu up. Confirm it has the same settings as Figure 22.

7) Click on the *Break emulation if this level true* box. This will break the emulation and record only that cycle which caused a write of 34 to memory location 5017. Click on OK to close the Trace Configuration window.

Figure 23



8) You must enter the value of 34 as 3400 because 5017 is an odd address, we are trying to trigger on a byte write (movb) and the ST10 is a 16 bit machine. The mask value of FF00 ensures only the 34 becomes the data of the qualifier. If you were using an address of 5016, the values used would be 0034 and 00FF. The leading zero of the 0034 can be discarded leaving 34. For 16 bit data qualifiers, these rules do not apply.

9) You should have a main menu that looks similar to Figure 17. Click on GO and the emulation will run. When the value at address 5017 becomes ASCII 4 (hex 34), the emulation will break and the instruction cycles will be displayed as in the trace window in Figure 24. The compressed feature is turned on.

10) Frame # -4 contains the write of 3400 to address 5017 with the movb instruction at address 442 ! Frame # -3 is the last instruction executed. R12 at this time contains 5018 and before this add #1, it contained 5017. The present value of R12 can be seen in the register window. R1 contains 0034 at this time.

Figure 24

Trace_1

Frame	Address	Destination	Source	Value	Data	Instr.		Symbol
-20	432:	AO2		2C	CO21	MOVBZ	R1,RL1	
-18	434:	FEOE	AO2	2C	F6F1OEFE	MOV	FEOEh,R1	
-17	438:				5BEE	DIVU	R14	
-7	43A:	AO2	FEOC	4	F2F1OCFE	MOV	R1,FEOCh	
-5	43E:	AO2	30	34	O7F23000	ADDB	RL1,#3Oh	
-4	442:	5017		3400	B92C	MOVB	[R12],RL1	
-3	444:		34	5018	O8C1	ADD	R12,#1h	

TRACE ONLY | got frames | Frames:416(-416:-1), Trig Count:0

11) Click on *Record this level in the trace buffer* in the window in Figure 23 and run the program again. Only the instruction that caused the trigger will show in the trace. You can focus on only that data you want.

12) In the Trace Configuration menu in the Level 1 tab, unselect B*reak emulation if this level true* so the emulation will not be stopped.

13) Click on the Reset and GO icons to restart emulation. Note you could also make this change "on-the-fly" without losing real-time performance.

14) Note the number of cycles recorded in the trace memory increasing as the value in 5017 equals 34. This information is shown at the bottom of the trace window. Stop the trace by clicking on the Trace icon and the cycles will be displayed as in Figure 25. Each cycle has a time stamp indication when it occurred.

15) Open the local menu and enable some more fields. Remember that not all cycles have been recorded: only those selected in the trigger qualifiers. This is only a small part of the trigger facilities of the ST10 emulator. This example does not begin to show the true capabilities of the emulator.

Notes: You can select the Advanced mode and begin to configure some extremely powerful triggers.

A trigger will turn the trace off at Frame # 0. The actual displayed value of the trigger point will vary a small amount due to the CPU pipeline effect.

Trace_1

Frame	Address	Destination	Source	Value	Data	Instr.		Symbol
-8	442:	5017		3400	B92C	MOVB	[R12],RL1	
-7	442:	5017		3400	B92C	MOVB	[R12],RL1	
-6	442:	5017		3400	B92C	MOVB	[R12],RL1	
-5	442:	5017		3400	B92C	MOVB	[R12],RL1	
-4	442:	5017		3400	B92C	MOVB	[R12],RL1	
-3	442:	5017		3400	B92C	MOVB	[R12],RL1	
-2	442:	5017		3400	B92C	MOVB	[R12],RL1	
-1	442:	5017		3400	B92C	MOVB	[R12],RL1	

TRACE ONLY | got frames | Frames:131071(-131071:-1), Trig Count:2

Figure 25

Level 7 Trigger Example:

1) Load the timer.abs example program.
2) Set the Level 7 Trigger as in Figures 26, 27 and 28. Figures 26 and 27 are accessible by right mouse clicks on the address and data areas of Figure 28. Set the Data Mask in Figure 28 to FF00. This will ensure the trigger looks only at the low byte. Make sure only Level 7 Trigger is active.
3) Open a Data window and set it to display 8 bit ASCII and from the address space TR Shadow. View address show or 5017. Open a trace window.
4) Start the emulation by clicking on the GO icon.
5) When address 5017 contains hex 34 (ASCII 4), the trace recording will stop. The Trace icon turns green.
6) The trigger point is located at Frame 0 in the Trace window. Scroll to this position if necessary. From the local menu selected by right clicking on the Trace window, select the Value field. Your Trace window will look similar to Figure 24. Note that value 34 was stored at 5017 by the movb instruction at Frame 0.
7) The trace memory is filled with the previous 128K cycles that preceded the trigger at Frame 0.

Figure 26

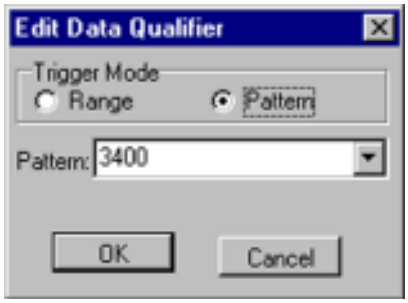
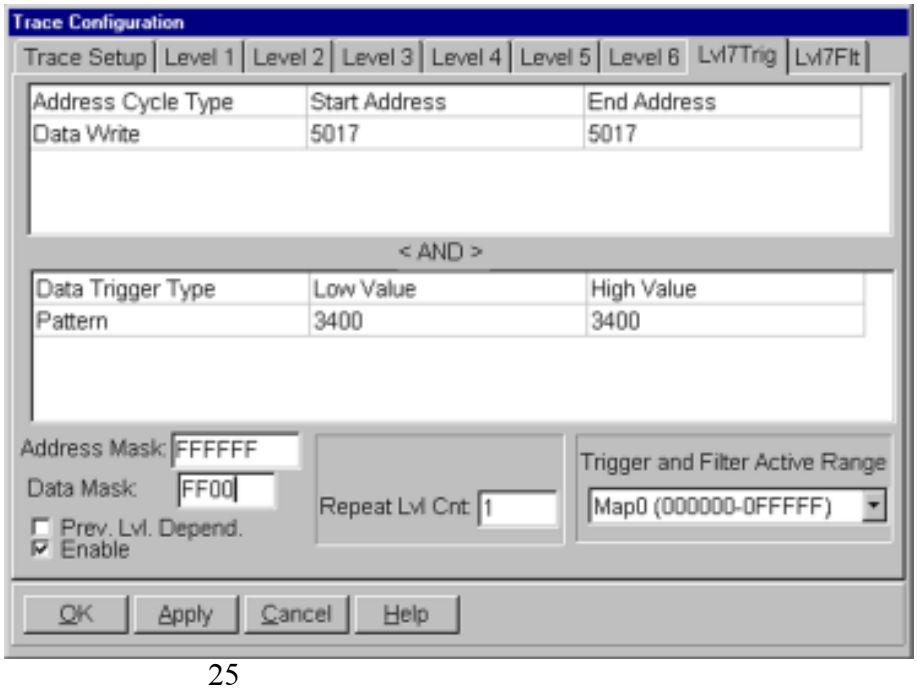


Figure 27

Figure 28



Level 7 Filter Example:

1) Load the timer.abs example program.
2) Set the Level 7 Filter as in Figure 29. Right mouse clicks to open up the local menus to enter the data. Make sure only Level 7 Filter is active so other triggers do not become events and confuse the situation.
3) Open a Data window and set it to display ASCII and from the address space Shadow or TR Shadow. Shadow is the Nohau Shadow memory and TR Shadow is the ST10 bondout Shadow memory. View address 5017 or type in *show*. Open a trace window. Arrange these windows for your convenience.
4) Start the emulation by clicking on the GO icon.
5) When address 5017 is written with a 34, this cycle will be recorded. The trace will not stop recording. Note that every 1 second or so the Frames Recorded field increases. The time is shown with the Relative Timestamp field selected in the trace local menu.
6) Stop the trace recording by clicking on the Trace icon. The trace window in Figure 30 will display a series of movb instructions. From the local menu selected by right clicking on the Trace window, select various fields to yield a display similar to Figure 30. You can also filter in Levels 1-6 by using the Record this level in the trace buffer attribute.
7) The trace memory is filled with only writes of 34 to memory location 5017. Note each instruction occurs 786 machine cycles from each other. You can also convert this to time in the local menu. Experiment with different settings of the local menu.
8) If you not put appropriate data qualifiers in the Data Trigger Type fields, all writes to 5017 will get recorded in the trace buffer.

Figure 29

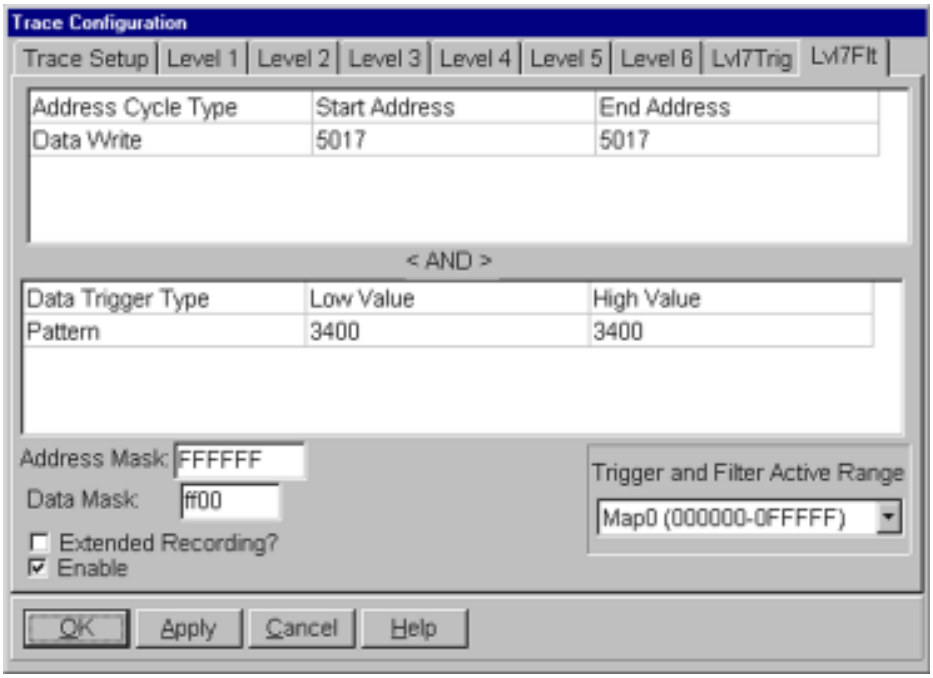
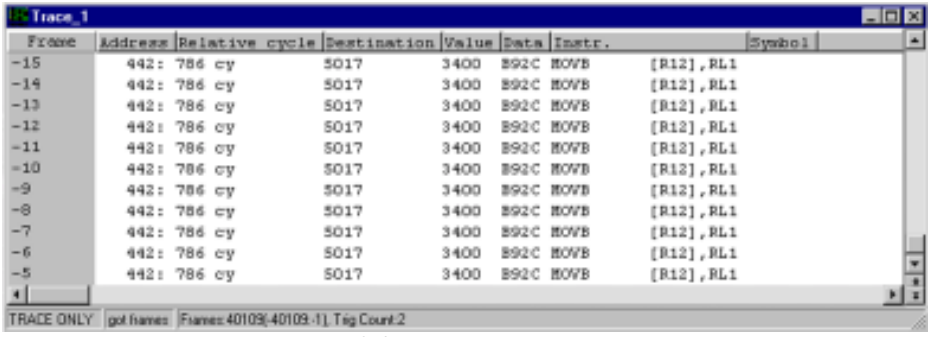


Figure 30



Group Breakpoints Examples

1) Set up Seehau and load the example program timer.abs.
2) Open a Source, data and Trace window and position them on the main menu to your preferences.
3) Configure the IP Group 1 as shown in Figures 31. Make sure the two enable boxes are checked. Figure 32 will be the configuration result. Click on OK to enter the qualifiers into the emulator.
4) Click on the GO icon to start the program.
5) The emulation will stop when a fetch of 442 occurs. This will be the last or second last instruction from the end in the trace window.
6) Experiment with different settings in the Edit Group dialog box shown in Figure 31.

Figure 31

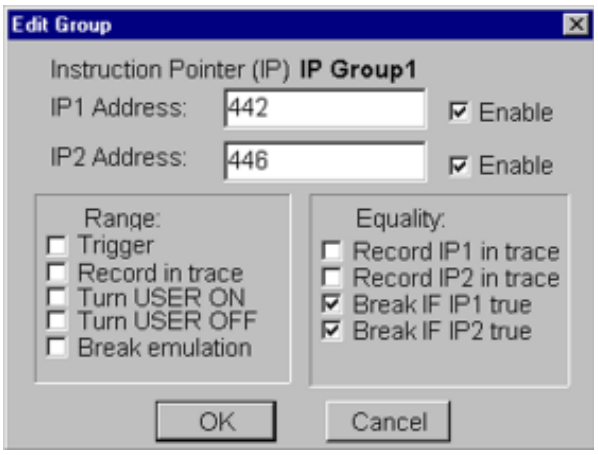
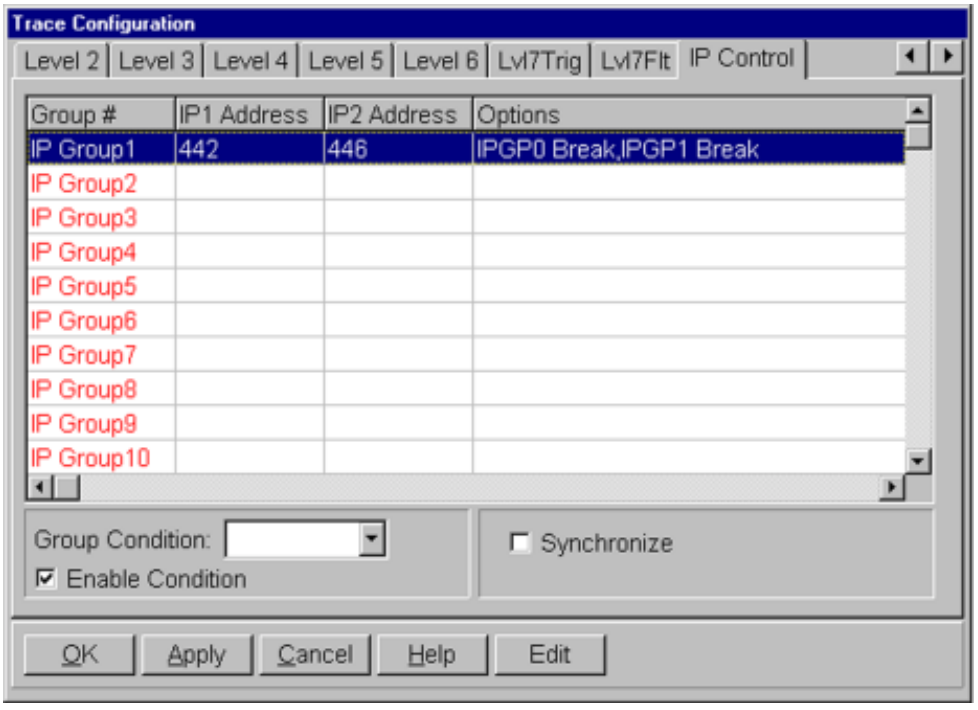


Figure 32



Trace Configuration Modification Conversion table

In November 1999, the Trace Configuration windows were modified to provide more intuitive English wording. The ST10 bondout mnemonics were previously used and caused some user difficulty. This list details the changes that were made. The macro commands are unchanged therefore use the old mnemonics. The old ST10 Quick Start can be obtained from the Nohau website for historical reference.

OLD	NEW
Window (Set)	Turn Trace recording ON
Window (Clear)	Turn Trace Recording OFF
Filter (Set)	Record this level in the trace buffer
Trigger (Set)	Trigger - stop trace and pulse Trigger Out
Break (Any level true)	Break emulation if this level is true
Break (All levels true)	Break emulation if all levels are true
BC_CAR	Repeat Counter reloaded
BC_CIC	Repeat Counter decrements on event ELSE cycle
BC_MCA	Repeat Counter not reloaded while in Event Mode
FRC_MBE	Make this Level always true (bypass mode)
PL_CMB	Previous Level must become true first
USER (Set)	Turn User Output ON
USER (Clear)	Turn User Output OFF
TIMEOUT 0 (Enable)	Enable Timeout 0
TIMEOUT 0 (disable)	Disable Timeout 0
TIMEOUT 1 (Enable)	Enable Timeout 1
TIMEOUT 1 (disable)	Disable Timeout 1
LC_CT0	Timeout 0 clears the Repeat Counter
LC_CT1	Timeout 1 clears the Repeat Counter
BC_MOD	Preset Address AND Data

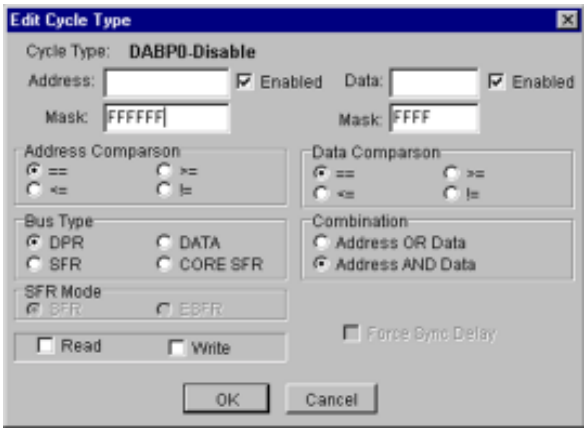
Notes:

No features are added or removed.
There is a direct one-to-one correlation between the old and new versions.
Customer Macros will not need modification.

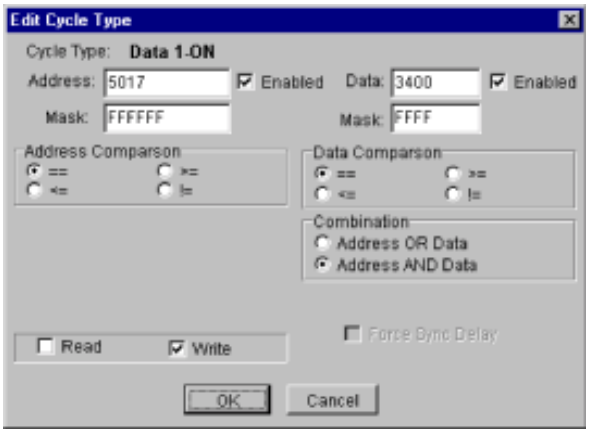
Changes to the EXTC: window:

OLD:	NEW:
No external source	no changes
Level 1 (memorized) *through* Level 6 (memorized)	Level 1 has occurred (latched) Level 6 has occurred (latched)
ILV1(current) < ILV1(prog)	CPU Int. level < Trigger Int. Level
MAC flag C carry generated *through* MAC flag SL: Limit flag	no changes no changes
MAC flag or combination	All 4 MAC flags ORd together
Timer-out 0 Timer-out 1	Timeout 0 = 3FF Timeout 1 = 3FF
External Trigger IN 0	External Trigger IN (J3)

Changes to the Edit Cycle Type Window Levels 1-6:



OLD:



NEW:

Conclusion

We hope this Application Note has helped you get acquainted with the Nohau ST10 emulator. If you have any problems please contact Nohau Technical Support or your local Nohau representative. For comments regarding the format of this document or any errors, please contact Robert Boys.

Sales@icetech.com
(800) 686-6428
www.icetech.com