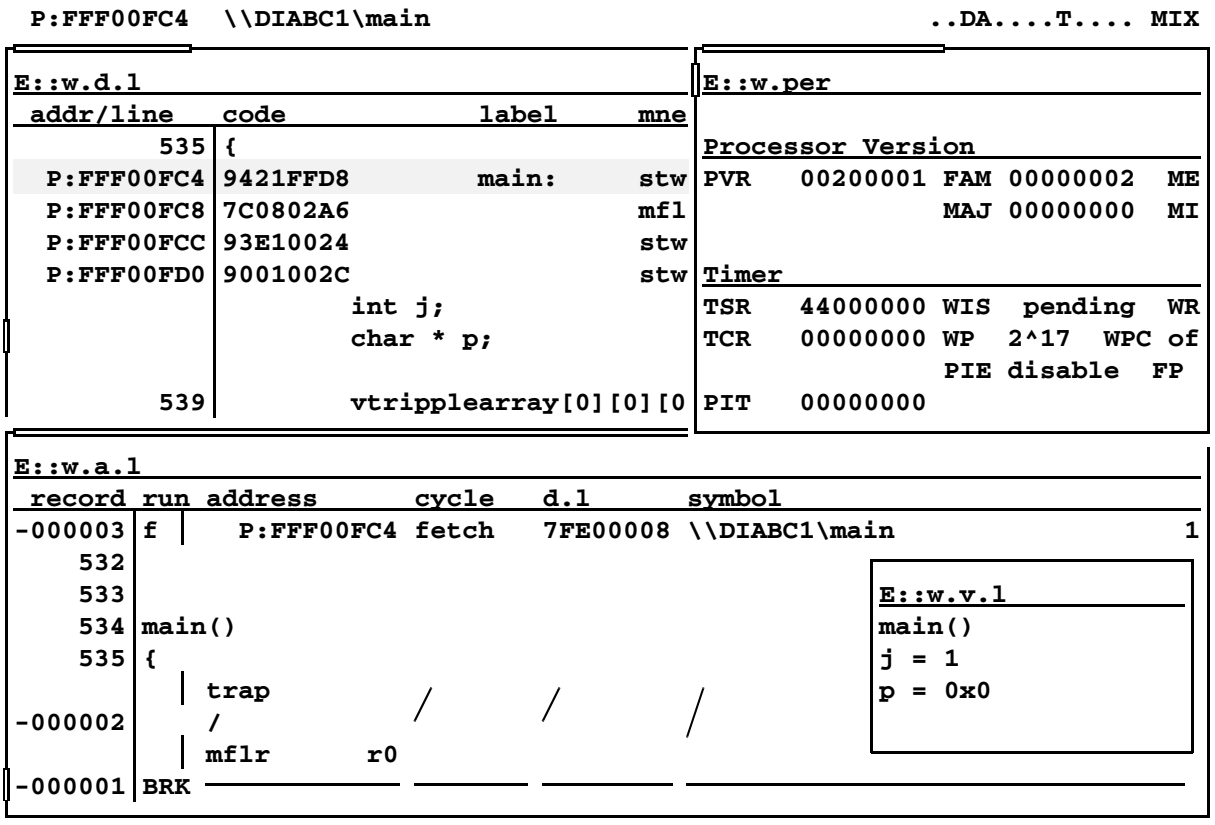
ICE Emulator for PowerPC

Version 06-Nov-2015



For general informations about the In-Circuit Debugger refer to the **"ICE User's Guide"** (ice_user.pdf). All general commands are described in **"IDE Reference Guide"** (ide_ref.pdf) and **"General Commands and Functions"**.

NOTE:	Do not connect or remove probe from target while target power is ON. Power up: Switch on emulator first, then target Power down: Switch off target first, then emulator

tbd.

Troubleshooting

Hang-Up

tbd.

Dualport Errors

tbd.

The ICE-PPC emulation head supports MPC500 and MPC800 series derivatives from Freescale Semiconductor and PPC400 series derivatives from IBM:

- PPC403GA
- MPC505
- MPC821
- MPC860

The adaption to different probes is done by changing the module. Modules support BGA or QFP versions, where applicable. The emulation frequency is up to 20MHz with 0 wait states and up to 28 MHz with 1 or more wait states. There is no significant speed difference to realtime because target systems in most cases use wait states and fast program loops are running from the cache. This leads to an average performance reduction of only about 10% using three wait states.

The probe uses a special emulation concept (active/passive emulation) to provide either emulation in realtime in the target or the advanced emulation features of Trace32 with reduced speed.

Therefore the probe contains three parts. The top level is the passive emulation module. It contains the drivers for addresses, data and ports, the control for the bus interface, the dualport and the BDM. The second level is the active CPU module, it contains the CPU, the interrupt-, reset-enable and the clock switches, the pull-up resistors for the CPU and the control of the switches and the buffers on the buffer module. The third level is the buffer module. It contains the address and data buffers between the CPU and the target. The modules are connected with the target connector for the CPU signals (e.g ET160) and additional an intermodul connector for the control signals. You can put these three modules one on the other.

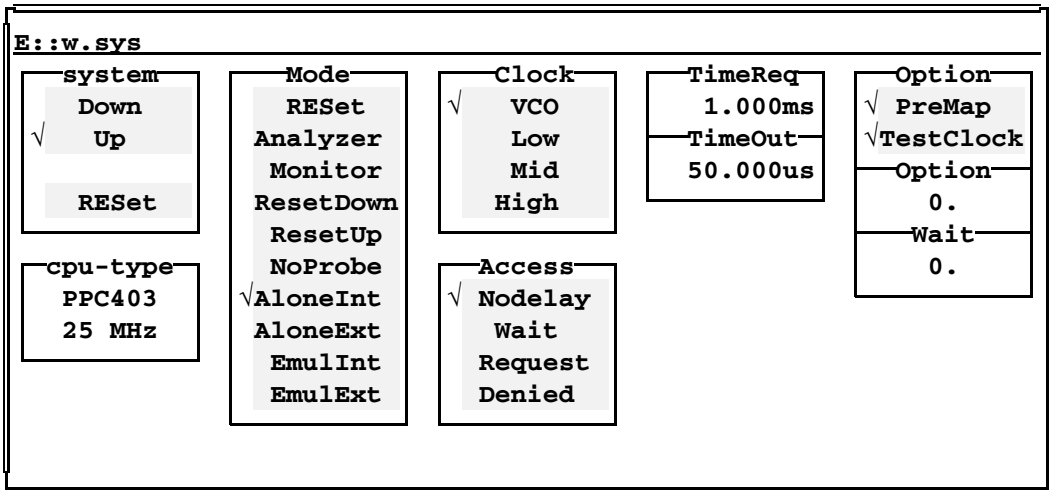
If you want to use all emulation features (Internal mode, internal mapping, internal clock) you need all three of the modules.

If you want more speed in your target memory, you can leave out the buffer module (the buffers have a few ns delay). The restriction now is, that you can map internal memory only if no buffer on your target is decoded at the same address, and that the synchronous breakpoints does not work with external memory. If your target memory is a ram, you can use software breakpoints instead.

If you have already soldered a CPU on your target, you can work in passive emulation. You need only the passive module. The CPU on your target is operated via the BDM port. Advantage is, that there is no time delay. Restrictions: Same as above and also no internal clock mode and no enable/disable of the interrupts and the reset lines.

An additional slot in the base modul offers upgrading with the port analyzer to get timing and state of the CPU ports.

We use a different system for numbering the address and data lines as it is used in the PowerPC descriptions. Our least significant bit is called D0 or A0, the MSB D31 or A31. Don't be confused, if you find some differences between your databook and perhaps our peripheral window.



The emulation head can stay in 6 modes. The modes are selected by the **SYStem.Up** or the **SYStem.Mode** command.

Format: **SYStem.Mode** <*mode*>

<*mode*>: **ResetDown**
ResetUp
AloneInt
AloneExt
EmulInt
EmulExt

Reset Down Target is down, all drivers are in tristate mode.

Reset Up Target has power, drivers are logically in inactive state, but not tristate.

Alone Internal Probe is running with internal clock, driver inactive. (Only with active module and buffer module.)

Alone External Probe is running with external clock, driver inactive. (Only with active module and buffer module.)

Emulation Internal Probe is running with internal clock, strobes to target are generated. (Only with active module, buffer module optional)

Emulation External Probe is running with external clock, strobes to target are activated. (Active module and buffer module optional)

In active mode, the power of the target is sensed and by switching down the target the emulator changes to **RESET** mode. The probe is not supplied by the target. When running without target, the target voltage is simulated by an internal pull-up resistor.

Format:	**SYStem.Clock** <*option*>
<*option*>:	**VCO** **High** **Mid** **Low**

VCO Variable frequency 1 … 35 MHz.

Low, Mid, High 2.5, 5.0 or 10.0 MHz.

Dualport Modes

Format:	**SYStem.Access** <*mode*>
<*mode*>:	**Request** **Denied**

Dualport access modes.

Request The CPU bus access is stopped by the bus request signals for dualport access.

Denied No dualport access is allowed while the realtime emulation is running.

Mapping and CS Setting of the MPC505, PPC403

The PowerPC controllers have bus interfaces, which allows the CPU to communicate with the external memory and peripherals without external logic. In the emulator, we use an epld to rebuild an address and data strobe out of the different chipselect lines of the CPU. For programming this epld, it is necessary for us to know, how the user has programmed the bus interface.

So, it is necessary for us to know these settings before delivering, to adapt this reconstruction epld to your target.

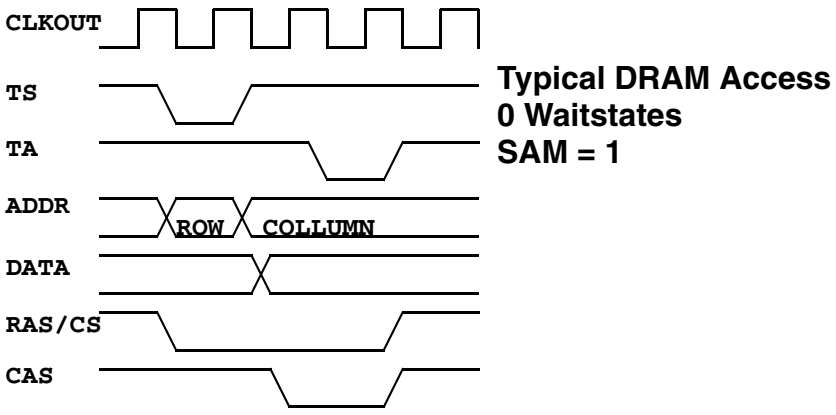
If the CS settings change during the project, it is possible to reprogram this epld. It is possible to reprogram it by sending a programming batch file to the customer.

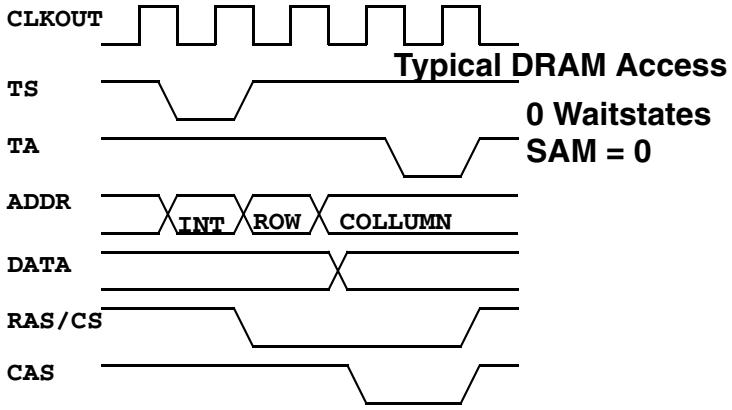
Mapping and CS setting of the MPC860/821

Normally, it is possible to map the user ram of an MPC860 emulator without any application specific epld. The following rules should enable you to make your own mapping:

1. Set the SYS.OPTION PreMapModule off

2. Set your CS registers in the way you want the mapping. If you use the UPMA or UPMB DRAM access, the start address multiplex bit (CSNT/SAM) in the option register of the CS must be 0. The emulator uses the first cycle of the memory access to latch the internal addresses. With this bit at 0, the first cycle of the DRAM access is not multiplexed, but shows the internal addresses on the address pins. For accessing your target memory, it is necessary to change the programming of the UPM RAM. It is recommended to use internal emulation memory instead of the external target DRAM. Then, you don't have to care for the UPM settings. Find more Information about this matter in the chapters External Bus Interface and Memory Controller of the user manual





3. Map workbenches wherever you need ram.

4. Map RAM wherever you need it. Map the bus size to the RAM location.

Example:

```
SYS.DOWN

MAP.RESET

SYS.O PMM OFF

SYS.M AI

d.s 0x2200110 %l 0x000800081
Set BR CS2
d.s 0x2200114 %l 0x0FC00800
Set OR CS2
MAP.PRE 0x0--0x0FFFFF

MAP.RAM 0x0--0x0FFFF
MAP 64 K
MAP.BUS32 0x0--0x0FFFF
32 Bit
MAP.I
internal
```

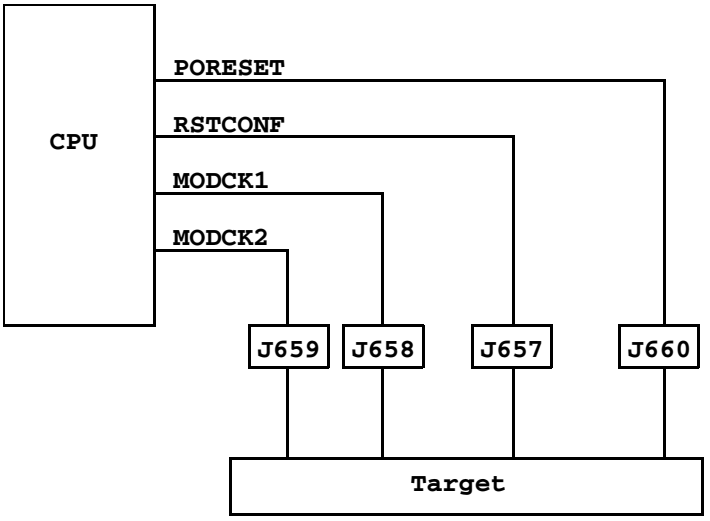
Jumper Settings of the MPC860/821 Probe

Some lines of the MPC860 CPU are connected with the target and can be disconnected by removing a jumper, if they cause problems to the target.

PORESET has a 4.7K pull-up and will be asserted with every system.up

RSTCONF has a 1K pull-down

MODCK has pull-up/pull-down as configured. Ref.: Adaption to different clock sources

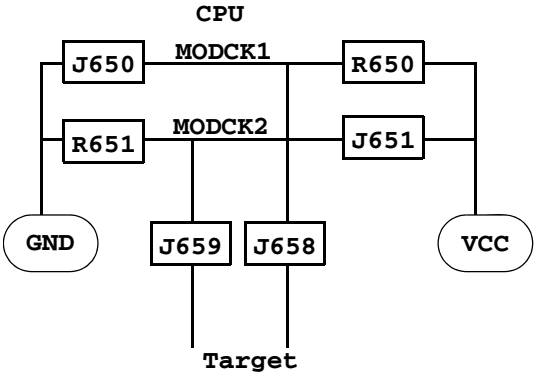


Look in Layout of the MPC860/821 probe for the physical location of the jumpers.

Adaption to different Clock Sources of the MPC860/821 Probe

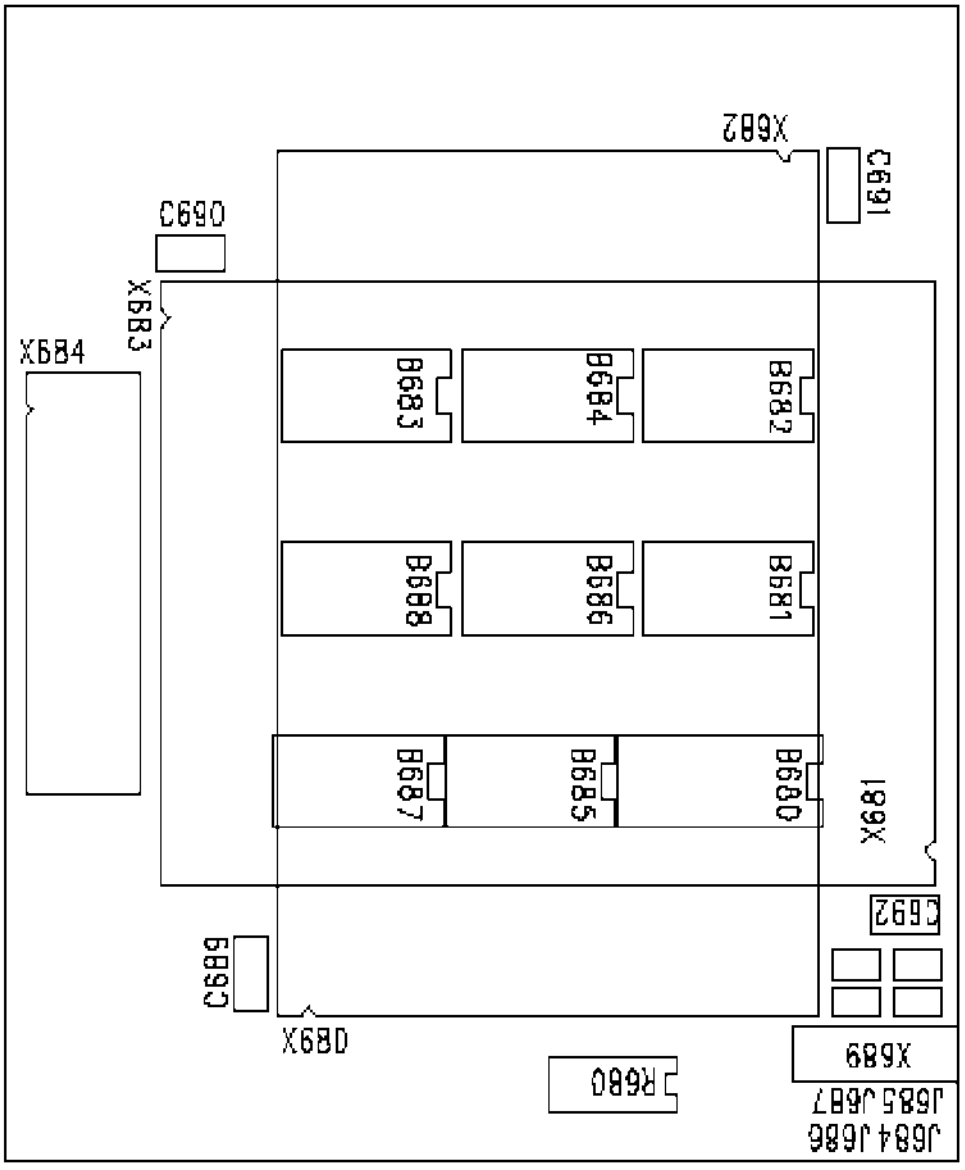
MPC860 Clock source selection is made by sampling the MODCK1 and MODCK2 pins during Power On Reset (POR). The POR is asserted during every sys.up command. The emulation pod has 4.7K pull-up at MODCK1 and 4.7K pull-down at MODCK2 as default. This is the 1:1 mode setting of the CPU clock. For adapting to different clock sources, you may change the pull-up/down resistors as needed.

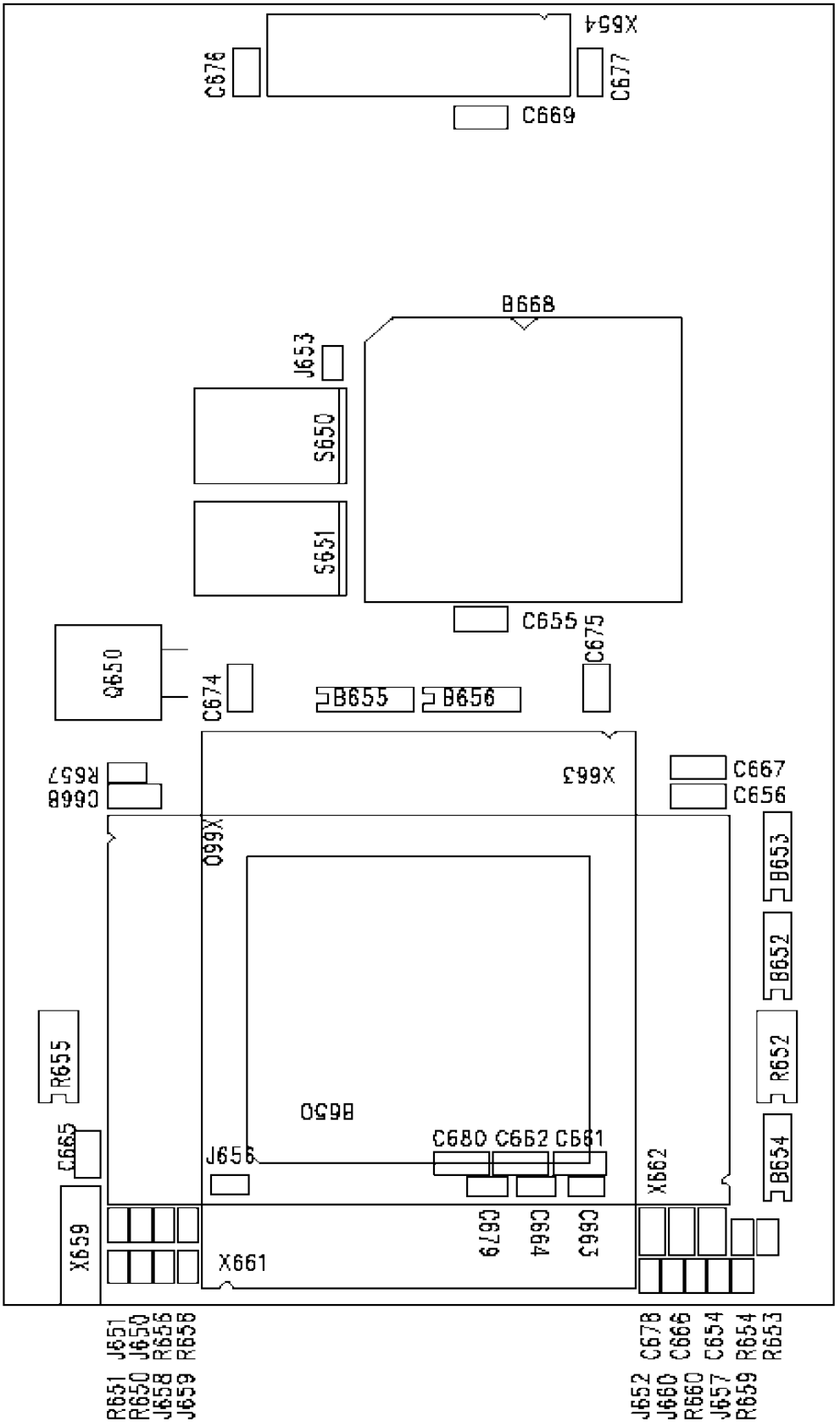
MODCK1	MODCK2	Default MF+1	SPLL Options
0	0	513	Normal Operation, PLL Enabled. Timing reference is freq (OSCM) = 32 kHz
0	1	5	Normal Operation, PLL Enabled. Timing reference is freq (OSCM) = 4 MHz
1	0	1	Normal Operation, PLL Enabled. 1:1 Mode F (clkout) = F (extclk)
1	1	5	Normal Operation, PLL Enabled. Timing reference is freq (OSCM) = 4 MHz



Look in Layout of the MPC860/821 probe for the physical location of the jumpers.

Layout of the MPC860/821 Probe





General Restrictions

Turbo Control (400 family)

Make sure, that you don't increase the debug clock without decreasing the internal wait states, when the TURBO option is enabled. If external wait states are used it is recommended to switch TURBO mode off.

Trace Information (400 family)

The emulator needs the trace information on the TS pins. You must switch the realtime debug mode in the input/output configuration register to bus.

Exception Routines (500/800 family)

The CPU handles the debug mode similar to an exception. Therefore stepping through an interrupt service routine is not possible, because the execution of the RFI instruction forces the CPU to exit from debug mode. Also modifications of SRR0 and SRR1 are ignored when exiting debug mode. If it is necessary to debug an exception routine, you are allowed to do the following things: If the CPU is not in a recoverable state (after jump into the exception routine) no breakpoints are allowed. When the software of the exception has saved the MSR and IP and set the RI bit of the MSR, the CPU is in recoverable state and one is allowed to break the routine. After this break, the old srr0 and srr1 registers, which contain the information about the state of the CPU before the exception are overwritten and lost. You can now step through the exception routine till the srr0 and srr1 registers are recalled from the stack. After this program line till the RFI instruction is reached, stepping (neither HLL nor ASM steps) or breaks are not allowed anymore, but it is possible to leave the exception routine with a go command.
While being in a non recoverable state, you can’t execute a go command

Clock Output (500/800 family)

The emulator needs the clockout frequency of the CPU, you must not switch off the clockout pin for power saving purposes.

Dualport

For the dualport access, it is necessary for the emulator to have the control of the bus between the cycles of the CPU. The emulator uses the normal bus arbitration signals to stop the CPU cycles. If you want to use dualport access, no device on your target may drive an active high signal on the bus, because then the emulator would produce an bus collision. Use pull-up resistors instead.

Format: **SYStem.Option PreMap [ON | OFF]**

The emulator can run in 24- and 32-Bit mode. If the upper address lines are not used by the target system, the pre-mapper should be switched off.

We call the most significant address A31, in difference to the PPC description.

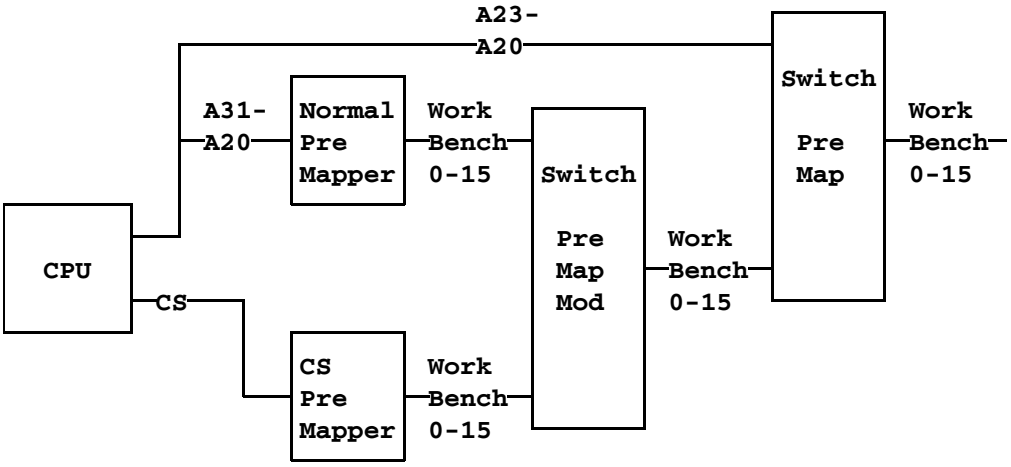
Bus width	**SYStem.Option PreMap**
A0--A24	OFF
A0--A32	ON

SYStem.Option PreMapMod Premapper mode

Format: **SYStem.Option PreMapMod [ON | OFF]**]

The emulator has two premapper. The first is the regular premapper, which uses the address lines 20 to 31. The second is the module premapper, which has defined the 16 workbenches to the CS signals and the address A20 (=A11 of the PowerPC) of the CPU. It is not as flexible a the regular premapper, but it is faster. Here, you can define your memory region for each workbench.

The sys.o pmm switches between the pre-mapper ram on the base and the project specific premap epld on the module.



For example:

	A20=0	A20=1
CS0	WB0	WB1
CS1	WB2	WB3
CS2	WB4	WB5
CS3	WB6	WB7
CS4	WB8	WB9
CS5	WB10	WB11
CS6	WB12	WB13
CS7	WB14	WB15

You want to use CS2 at the address 0ff00000. If you want to use the module premapper type:

```
sys.o pmm on
map.pre 0x0ff00000++0x0fffff 4.
map …
```

Now your map.pre window shows following:

logical Workbench	Address	physical Workbench
1	0FF00000--0FFFFFFF	4

If you want to use the regular premapper type:

```
sys.o pmm off
map.pre 0x0ff00000++0x0fffff
map …
```

Now your map.pre window shows following:

logical Workbench	Address	physical Workbench
1	0FF00000--0FFFFFFF	0

SYStem.Option TestClock

Clock test

Format: **SYStem.Option TestClock [ON | OFF]**

If this option is on, the emulator tests if there is a clock output of the CPU.

SYStem.Option BreakMask — Break mask

Format: **SYStem.Option BreakMask [ON | OFF]**

Only MPC860, MPC821, MPC505

The cpu handles debug events similar to exceptions. When a debug event (normally a break) OR a exception occurs, the cpu copies the msr into srr1 and the ip into srr0. This means, that after an exception occurred, the old values of ip and msr are as backup in the srr0 and srr1 registers. If now a break happens, these values will be overwritten by the new msr and ip values. So, it is possible to return to the exception routine, but not to the main program. The status after the start of the exception routine is called non recoverable state.

If you want to break in a non recoverable state, you must switch the option BreakMask to on.

SYStem.Option FREEZE — Timer freeze modes

Format: **SYStem.Option FREEZE [ON | OFF]**

If this option is on, the internal timer/counter are frozen when being in debug mode.

SYStem.Option VSYNC — Synch trace signals

Format: **SYStem.Option VSYNC [ON | OFF]**

Only MPC860, MPC821, MPC505

The emulator has the possibility to trace the flowtrace signals of the cpu with each clock cycle. With this trace, it is possible to reconstruct the instruction flow of the cpu, even if the cpu runs in the internal cache. To reconstruct this flow, it is necessary that the cpu makes a show cycle after each indirect branch (See register setting of the ICTRL register in your cpu manual) and, that the cpu makes one show cycle after the half of the clock trace.

The option VSYNC generates a VSYNC command to the cpu every 32000 clock cycles to force the cpu to make a show cycle.

SYStem.Option CFLUSH — Instruction cache flush

Format: **SYStem.Option CFLUSH [ON | OFF]**

Only MPC860, MPC821, MPC505

If you use the internal instruction cache, it is necessary to flush the cache before every go or hll step. Option CFLUSH enables the cache flush software before each jump in.

Warning: Problems can occur when the LCD driver of the MPC821 is active!!!

SYStem.Option ONCE — On-circuit emulation

Format: **SYStem.Option ONCE [ON | OFF]**

Only MPC860, MPC821

If you use the target connection via samtec connectors, and leave the CPU on the target, you can switch the CPU on the target to HI-Z state. For this option, it is necessary that the BDM/JTAG pins of the target CPU are in BDM mode after reset (default), and that the TRST pin is pulled to high with a resistor (1K - 10K).

SYStem.Option BASE — Peripheral base address

Format: **SYStem.Option BASE [<Value>]**

Only MPC860, MPC821

The internal peripherals of the cpu can be mapped at different places. Sys.o base defines the base address of the peripheral window. This option must be set before the peripheral window is activated. If it is changed later, you must reprogram the peripheral window with the command per.rp.

SYStem.Option RESETCONF — Reset configuration

Format: **SYStem.Option RESETCONF [<Value>]**

Only MPC860, MPC821

After HRESET is released, the reset configuration word is sampled from the data bus. With this option, you can define your reset configuration. The DBGC value is always 0x3 and the DBPC value is always 0.

SYStem.Option IBUS — IBUS control

Format: **SYStem.Option IBUS [<Value>]**

Only MPC860, MPC821

With this option, you can set the instruction fetch show cycle and serialize control bits of the IBUS support control register.

SYStem.Option ICFLUSH — Internal instruction cache flush

Format: **SYStem.Option ICFLUSH [ON | OFF]**

If you use the internal instruction cache, it is necessary to flush the cache before every go or hll step. Option CFLUSH enables the cache flush software before each jump in.

Warning: Problems can occur when the LCD driver of the MPC821 is active!!!

SYStem.Option ICREAD — Instruction cache read

Format: **SYStem.Option ICREAD [ON | OFF]**

If this option is switched on, data dump considers the valid cache lines of the instruction cache. This can be different to the external memory.

SYStem.Option DCREAD — Date cache read

Format: **SYStem.Option DCREAD [ON | OFF]**

If this option is switched on, data dump considers the valid cache lines of the data cache. This can be different to the external memory.

SYStem.Option WATCHDOG — Disable watchdog

Format: **SYStem.Option WATCHDOG [ON | OFF]**

If this option is switched off, the watchdog timer of the CPU is disabled after the sys.up.

SYStem.Option BRKNOMSK — tbd.

Format: **SYStem.Option BRKNOMSK [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

SYStem.Option FlowTrace

tbd.

Format: **SYStem.Option FlowTrace [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

SYStem.Option ResetExt

tbd.

Format: **SYStem.Option ResetExt [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

SYStem.Option ResetMode

tbd.

Format: **SYStem.Option ResetMode [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

SYStem.Option SCRATCH

tbd.

Format: **SYStem.Option SCRATCH [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

SYStem.Option TURBO

tbd.

Format: **SYStem.Option TURBO [ON | OFF]**

ON tbd.

OFF tbd.

tbd.

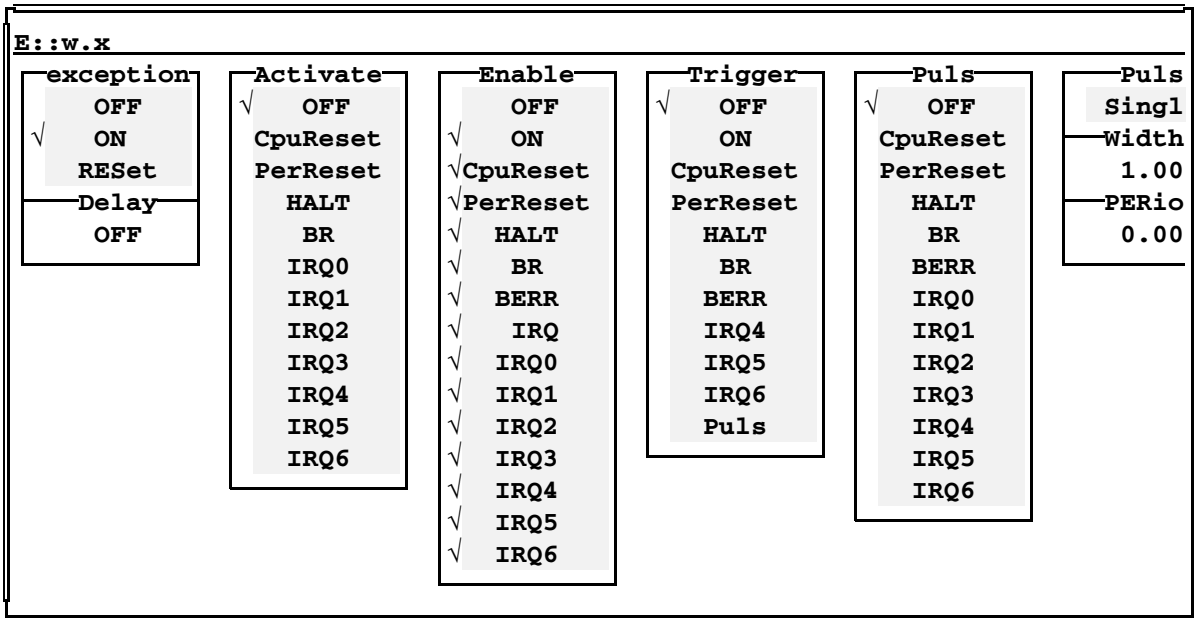
SYStem.Option Wait

System wait states

Format: **SYStem.Option Wait** [<*wait_states*>]

Number of additional system wait states.

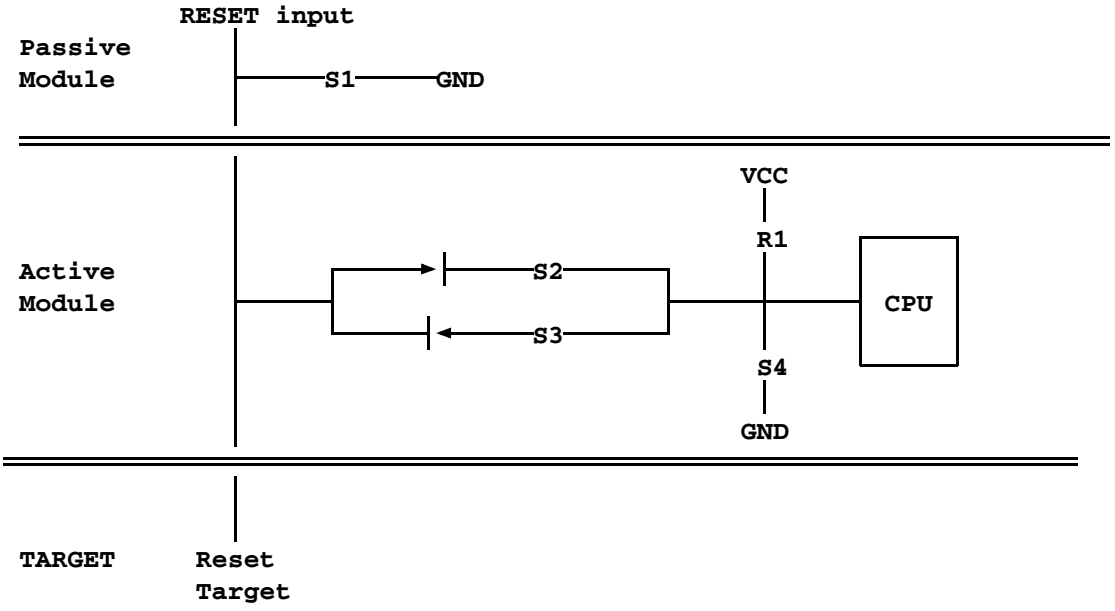
Exception Control



This menu chart is from the PPC403. It may be different for other CPUs.

Schematics

The reset line (input and output) of the active module is controlled by a bridge with analog switches and diodes (PPC400).




```
R1 = 1K

S1   Reset Passive   X.Activate ResetP
                     X.Puls ResetP

S2   Reset Out       X.Enable Resetout

S3   Reset In        X.Enable Reset

S4   Reset CPU       X.Activate ResetC
                     X.Puls     ResetC
```

Format:	**eXception.Activate CINT [ON | OFF]**
Format:	**eXception.Activate HRESET [ON | OFF]**
Format:	**eXception.Activate SRESET [ON | OFF]**
Format:	**eXception.Activate RESET [ON | OFF]**
Format:	**eXception.Activate RESETC [ON | OFF]**
Format:	**eXception.Activate RESETH [ON | OFF]**
Format:	**eXception.Activate RESETC [ON | OFF]**
Format:	**eXception.Activate INT0..INT4 [ON | OFF]**
Format:	**eXception.Activate IRG0..IRQ7 [ON | OFF]**
Format:	**eXception.Activate OFF**

CINT	Activates the CINT line.
HRESET	Activates the HRESET line.
SRESET	Activates the SRESET line.
RESET	Activates the RESET line.
RESETC	Activates the RESET line.
RESETO	Activates the RESET line.
RESETT	Activates the RESET line.
INT0	Activates the INT0 line.
INT1	Activates the INT1 line.
INT2	Activates the INT2 line.
INT3	Activates the INT3 line.

INT4	Activates the INT4 line.
IRQ0	Activates the IRQ0 line.
IRQ1	Activates the IRQ1 line.
IRQ2	Activates the IRQ2 line.
IRQ3	Activates the IRQ0 line.
IRQ4	Activates the IRQ1 line.
IRQ5	Activates the IRQ2 line.
IRQ6	Activates the IRQ0 line.
IRQ7	Activates the IRQ1 line.
OFF	No activation of any exception line.

Format:	**eXception.Enable CINT [ON | OFF]**
Format:	**eXception.Enable HRESET [ON | OFF]**
Format:	**eXception.Enable SRESET [ON | OFF]**
Format:	**eXception.Enable RESET [ON | OFF]**
Format:	**eXception.Enable RESETO [ON | OFF]**
Format:	**eXception.Enable INT [ON | OFF]**
Format:	**eXception.Enable INT0 ..INT4 [ON | OFF]**
Format:	**eXception.Enable IRQ [ON | OFF]**
Format:	**eXception.Enable IRQ0 .. IRQ7 [ON | OFF]**
Format:	**eXception.Enable OFF**
Format:	**eXception.Enable ON**

CINT	Enables the CINT line.
HRESET	Enables the HRESET line.
SRESET	Enables the SRESET line.
RESET	Enables the RESET line.
RESETO	Enables the RESETO line.
INT	Enables the INT line.
INT0	Enables the INT0 line.
INT1	Enables the INT1 line.
INT2	Enables the INT2 line.
INT3	Enables the INT3 line.

INT4	Enables the INT4 line.
IRQ	Enables the IRQ line.
IRQ0	Enables the IRQ0 line.
IRQ1	Enables the IRQ1 line.
IRQ2	Enables the IRQ2 line.
IRQ3	Enables the IRQ1 line.
IRQ4	Enables the IRQ2 line.
IRQ5	Enables the IRQ1 line.
IRQ6	Enables the IRQ2 line.
IRQ7	Enables the IRQ2 line.
ON	Enables all exception line.
OFF	Disables all exception lines.

eXception.Pulse — Stimulate exception

Format:	**eXception.Pulse CINT**
Format:	**eXception.Pulse HRESET**
Format:	**eXception.Pulse SRESET**
Format:	**eXception.Pulse RESET**
Format:	**eXception.Pulse RESETO**
Format:	**eXception.Pulse RESETC**
Format:	**eXception.Pulse RESETT**

Format:	**eXception.Pulse INT0 .. INT7**
Format:	**eXception.Pulse IRQ0 .. IRQ7**
Format:	**eXception.Pulse OFF**

CINT	Stimulate CINT line.
HRESET	Stimulate HRESET line.
SRESET	Stimulate SRESET line.
RESET	Stimulate RESET line.
RESETC	Stimulate RESETC line.
RESETO	Stimulate RESETO line.
RESETT	Stimulate RESETT line.
INT0	Stimulate INT0 line.
INT1	Stimulate INT1 line.
INT2	Stimulate INT2 line.
INT3	Stimulate INT3 line.
INT4	Stimulate INT4 line.
IRQ0	Stimulate IRQ0 line.
IRQ1	Stimulate IRQ1 line.
IRQ2	Stimulate IRQ2 line.
IRQ3	Stimulate IRQ3 line.
IRQ4	Stimulate IRQ4 line.
IRQ5	Stimulate IRQ4 line.
IRQ6	Stimulate IRQ4 line.
IRQ7	Stimulate IRQ4 line.
OFF	No stimulation on any exception line.

MAP.BUS — Bus width mapping

Format: **MAP.BUS8** [<*range*>]
MAP.BUS16 [<*range*>]
MAP.BUS32 [<*range*>]
MAP.BUSEXT [<*range*>]

Every block in the address space of the CPU has either an 8, 16 or 32 bit bus width. The emulator breakpoint and trace system need this information in realtime in order to work correctly. The mapper must be set for all ranges, where internal bus width setting is used.

```
map.bus8  0x0--0x0fffff          ; maps first 1 MB block for 8 bit

map.busext                       ; remaps all to external definition
```

The **MAP.RESet** command sets the bus width definition to external.

MMU.TLB — Display MMU TLB entries

Format: **MMU.TLB** <*tlb*>

<*tlb*>: **IMMU**
DMMU

Displays a table of all MMU TLB entries of the selected TLB table.

MMU.TLBSCAN — Load MMU TLB entries

Format: **MMU.TLBSCAN**
MMU.TLBSCAN <*tlb*>

<*tlb*>: **IMMU**
DMMU

Loads the TLB table entries from the CPU to the debugger internal MMU table.

Memory Class	Description
P	Program
D	Data
SPR	Special Purpose Register
DCR	Device Control Register (400)

P: and D: This storage classes operate on the same physical memory. They are only used to be compatible with other emulation probes. CPU internal registers and memory may not be accessed dualported.

Keywords for the Trigger Unit

Input Event	Meaning	Analyzer Hardware			
		ECC8	HAC	HA120	SA120
BURST				X	X
BUS8	8 bit bus access			X	X
BUS16	16 bit bus access			X	X
BUS32	32 bit bus access			X	X
DATA	Data access		X	X	X
DMA	DMA cycle		X	X	X
FETCH	Opcode fetch		X	X	X
IO	IO cycle		X	X	X
Read	Read cycle		X	X	X
Write	Write Cycle		X	X	X

For not CPU-specific keywords, see **non-declarable input variables** in **"ICE/FIRE Analyzer Trigger Unit Programming Guide"** (analyzer_prog.pdf).

Keywords for the Display

WAIT	Wait for interrupt, normally not sampled
PA0..PA7	Port A

Keywords for the Port Analyzer

PPC403GA

P.BUSERR
P.BUSREQ
P.CINT
P.DMA0--P.DMA3
P.DMAR0--P.DMAR3
P.DSR
P.DTR
P.EOT0--P.EOT3
P.ERROR
P.HALT
P.HOLDACK
P.HOLDREQ
P.INT0--P.INT4
P.OE
P.READY
P.SCLK
P.TCLK
P.TS0--P.TS6
P.XMITD
P.Z0--P.Z9

MPC505

P.A0--P.A7
P.B0--P.B7
P.I0--P.I7
P.J1--P.J7
P.K0--P.K5
P.L2--P.L7
P.M3--P.M7
P.Q0--P.Q6
P.Z0--P.Z3
P.DS
P.TCK
P.TMS
P.TRST
P.TDI
P.TDO

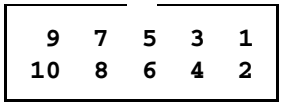
MPC860/MPC821: Ports are Multiplexed on the Module, Switch with P.MUX A, P.MUX B. The port analyzer must be switched before tracing the ports

MPC860 (A)	MPC860(B)
P.IRQ0--P.IRQ7	P.PA00--P.PA15
P.CS0--P.CS7	P.PB14--P.PB31
P.WE0--P.WE3	P.PC04--P.PC15
P.GPLA0--P.GPLA5	P.PD03--P.PD15
P.GPLB4	
P.IPA0--P.IPA7	
P.IPB0--P.IPB7	
P.ALEB	
P.BSA0--P.BSA3	
P.OP0--P.OP1	
P.MODCK1--P.MODCK2	
P.DP0--P.DP3	
P.HRESET	
P.SRESET	
P.PORESET	
P.Z0--P.Z4	P.Z0--P.Z4

Additional Trace Channels

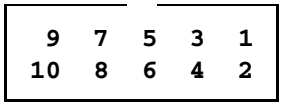
Not used trace channels on Port Analyzer are connected to pins placed on the emulation module.

Module PPC403GA

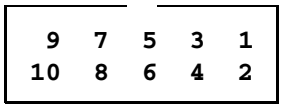


1	Port.Z0
2	Port.Z1
3	Port.Z2
4	Port.Z3
5	Port.Z4
6	Port.Z5
7	Port.Z6
8	Port.Z7
9	Port.Z8
10	Port.Z9

Module MPC505



1	Gnd
2	Gnd
3	Port.Z0
4	Port.Z1
5	Gnd
6	Gnd
7	Port.Z2
8	Port.Z3
9	GND
10	GND



1	Port.Z0
2	Port.Z1
3	Port.Z2
4	Port.Z3
5	Port.Z4
6	Gnd
7	Gnd
8	Gnd
9	Gnd
10	Gnd

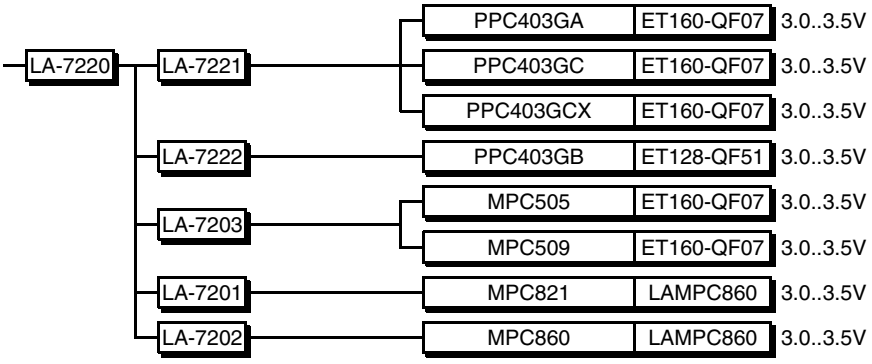
Language	Compiler	Company	Option	Comment
ADA	GNAT	Free Software Foundation, Inc.	ELF/DWARF	
C	CXPPC	Cosmic Software	ELF/DWARF	
C	CC	Freescale Semiconductor, Inc.	XCOFF	
C	XCC-V	GAIO Technology Co., Ltd.	SAUF	
C	GREEN-HILLS-C	Greenhills Software Inc.	ELF/DWARF	
C	GCC	HighTec EDV-Systeme GmbH	ELF/DWARF	
C	MCCPPC	Mentor Graphics Corporation	ELF/DWARF	
C	ULTRA-C	Radisys Inc.	ROF	
C	HIGH-C	Synopsys, Inc	ELF/DWARF	
C	DCPPC	TASKING	ELF/DWARF	
C	D-CC	Wind River Systems	IEEE	
C	D-CC	Wind River Systems	COFF	
C	D-CC	Wind River Systems	ELF/DWARF	
C++	GCC	Free Software Foundation, Inc.	ELF/DWARF	
C++	GREEN-HILLS-C++	Greenhills Software Inc.	ELF/DWARF	
C++	CCCPPC	Mentor Graphics Corporation	ELF/DWARF	
C++	MSVC	Microsoft Corporation	EXE/CV5	WindowsCE
C++	HIGH-C++	Synopsys, Inc	ELF/DWARF	
C++	D-C++	Wind River Systems	ELF/DWARF	
C++	GCCPPC	Wind River Systems	ELF/STABS	
C/C++	CODEWARRIOR	Freescale Semiconductor, Inc.	ELF/DWARF	
GCC	GCC	Free Software Foundation, Inc.	ELF/DWARF	
JAVA	FASTJ	Wind River Systems	ELF/DWARF	

CPU	Tool	Company	Host
ALL	ADENEO	Adeneo Embedded	
ALL	X-TOOLS / X32	blue river software GmbH	Windows
ALL	CODEWRIGHT	Borland Software Corporation	Windows
ALL	CODE CONFIDENCE TOOLS	Code Confidence Ltd	Windows
ALL	CODE CONFIDENCE TOOLS	Code Confidence Ltd	Linux
ALL	EASYCODE	EASYCODE GmbH	Windows
ALL	ECLIPSE	Eclipse Foundation, Inc	Windows
ALL	RHAPSODY IN MICROC	IBM Corp.	Windows
ALL	RHAPSODY IN C++	IBM Corp.	Windows
ALL	CHRONVIEW	Inchron GmbH	Windows
ALL	LDRA TOOL SUITE	LDRA Technology, Inc.	Windows
ALL	UML DEBUGGER	LieberLieber Software GmbH	Windows
ALL	ATTOL TOOLS	MicroMax Inc.	Windows
ALL	VISUAL BASIC INTERFACE	Microsoft Corporation	Windows
ALL	LABVIEW	NATIONAL INSTRUMENTS Corporation	Windows
ALL	CODE::BLOCKS	Open Source	-
ALL	C++TEST	Parasoft	Windows
ALL	RAPITIME	Rapita Systems Ltd.	Windows
ALL	DA-C	RistanCASE	Windows
ALL	TRACEANALYZER	Symtavision GmbH	Windows
ALL	SIMULINK	The MathWorks Inc.	Windows
ALL	TA INSPECTOR	Timing Architects GmbH	Windows
ALL	UNDODB	Undo Software	Linux
ALL	VECTORCAST	Vector Software	Windows
ALL	WINDOWS CE PLATF. BUILDER	Windows	Windows
POWERPC	GR228X IC-TESTSYSTEME	Battefeld GmbH	Windows
POWERPC	OSE ILLUMINATOR	Enea OSE Systems	Windows
POWERPC	DIAB RTA SUITE	Wind River Systems	Windows

Realtime operation Systems

Name	Company	Comment
AMX	KadakProducts Ltd.	
ChorusOS	Oracle Corporation	
CMX-RTX	CMX Systems Inc.	
DEOS	DDC-I, Inc.	implemented by DDC-I
ECOS	eCosCentric Limited	1.3, 2.0 and 3.0
Elektrobit tresos	Elektrobit Automotive GmbH	via ORTI
ERCOSEK	ETAS GmbH	via ORTI
Erika	Evidence	via ORTI
FreeRTOS	Freeware I	v7
Linux	-	Kernel Version 2.4 and 2.6, 3.x, 4.x
Linux	MontaVista Software, LLC	3.0, 3.1, 4.0, 5.0
LynxOS	LynuxWorks Inc.	3.1.0, 3.1.0a, 4.0
MQX	Freescale Semiconductor, Inc.	3.x and 4.x
MQX	Synopsys, Inc	2.40 and 2.50
NetBSD	-	
NORTi	MISPO Co. Ltd.	
Nucleus PLUS	Mentor Graphics Corporation	
OS-9	Radisys Inc.	
OSE Delta	Enea OSE Systems	4.x and 5.x
OSEK	-	via ORTI
OSEKturbo	Freescale Semiconductor, Inc.	via ORTI/former MetrowerksOSEK
PikeOS	Sysgo AG	
ProOSEK	Elektrobit Automotive GmbH	via ORTI
pSOS+	Wind River Systems	2.1 to 2.5, 3.0, with TRACE32
QNX	QNX Software Systems	6.0 to 6.5.0
RTEMS	RTEMS	4.10
RTXC 3.2	Quadros Systems Inc.	
RTXC Quadros	Quadros Systems Inc.	
Sciopta	Sciopta	
SMX	Micro Digital Inc.	3.4 to 4.0
ThreadX	Express Logic Inc.	3.0, 4.0, 5.0
uC/OS-II	Micrium Inc.	2.0 to 2.92
uITRON	-	HI7000, RX4000, NORTi,PrKernel
VRTXsa	Mentor Graphics Corporation	
VxWorks	Wind River Systems	5.x to 7.x

Module Overview



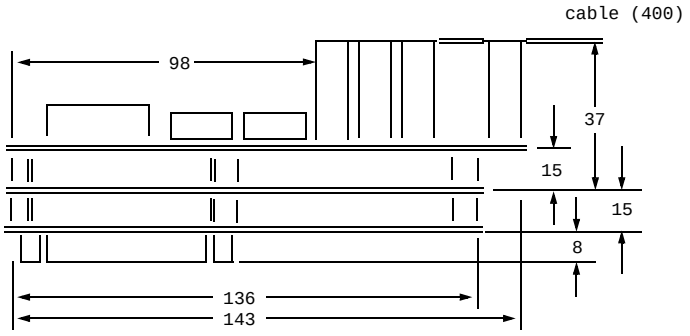
Order No.	Code	Text
AI-7206	BGA272-EXTENSION	BGA272-Adapter Extension
AI-7207	BGA357-AI-MALE	Advanced Interconnect male-male-block for MPC
AI-7208	BGA357-AI-SOCKET	Advanced Interconnect socket for MPC860/880
AI-7209	BGA357-EXTENSION	BGA357-Adapter Extension
AI-9546	BGA256-AI-SOCKET	Advanced Interconnect socket for MPC850
AI-9549	BGA272-AI-SOCKET	Advanced Interconnect socket for MPC555
AI-9664	BGA388-AI-SOCKET	Advanced Interconnect socket for MPC561/563 A
AI-9667	BGA388-ADAPTER	Advanced Interconnect BGA388 Adapter for CPU
AI-9672	BGA388-MALE-MALE-28	BGA388 Male-Male Connector 0.28mm Pin
AI-9673	BGA388-MALE-MALE-22	BGA388 Male-Male Connector 0.20mm Pin
LA-7210	BGA357-ETEC-MPC860	Emulation adapter for E-TEC socket for MPC860
LA-7211	CONNECTOR-ADS-MPC860	Emulation adapter for ADS board
LA-7213	BGA357-AI-MPC860	Emulation adapter for AI socket for MPC860
LA-7214	BGA357-ETEC-SOCKET	E-TEC socket for MPC860 (SMD)
LA-7215	A-MPC860-BOTTOM	Bottom Side Target Adapter for MPC860
LA-7217	CON.-FADS-MPC860	Emulation adapter for FADS board
LA-7218	BGA357-ETEC-SOCKET-T	E-TEC socket for MPC860 (through hole)
LA-7907	TCON320-BGA357-PPC	Emulation adap. f. TCON320 to BGA357-MPC880
LA-9545	BGA256-AI-MPC850/PPC	Emulation adapter for AI socket for MPC850
LA-9548	BGA272-AI-MPC555	Emulation adapter for AI socket for MPC555
LA-9660	TCON200-MPC823-AMC	Converter TCON 200 to AMC Footprint 823
LA-9661	TCON240-AI-MPC555	Emulation adap. from TCON240 to BGA272-MPC555
LA-9666	TCON320-AI-MPC56X	Emulation adap. from TCON320 to BGA388-MPC56x
Additional Options		
LA-7216	BGA357-CPU-ADAPTER	CPU Test Adapter for BGA357 (MPC860)

Operating Voltage

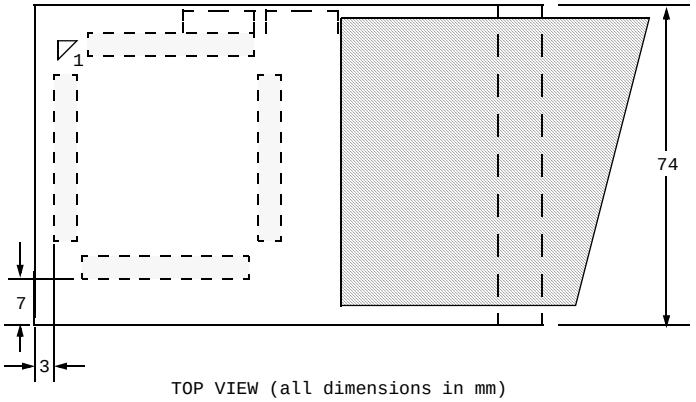
tbd

Dimension

LA-7222 M-PPC403GB-QFP



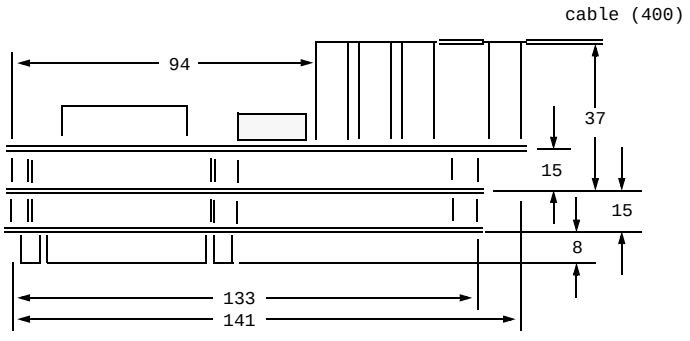
SIDE VIEW



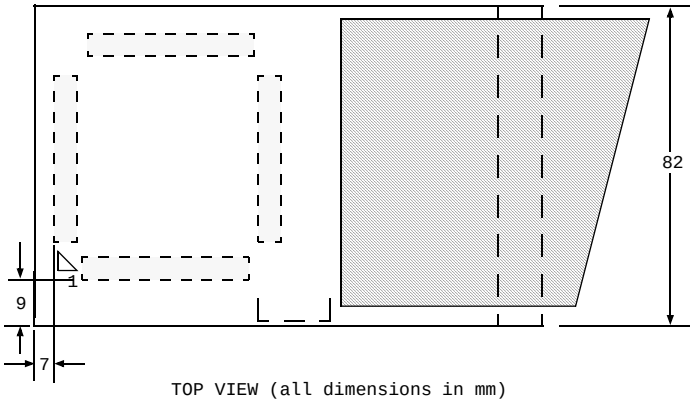
TOP VIEW (all dimensions in mm)

Dimension

LA-7203 M-MPC505-QFP



SIDE VIEW

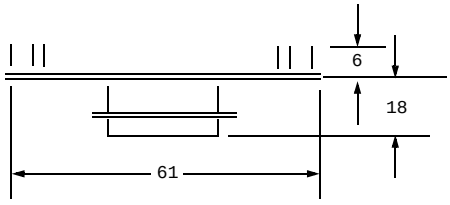


TOP VIEW (all dimensions in mm)

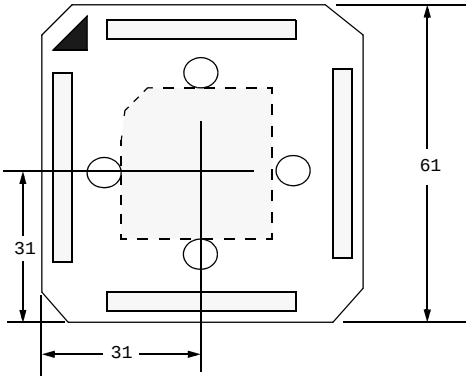
LA-7213 BGA357-AI-MPC860

Dimension

LA-7210 BGA357-ETEC-MPC860



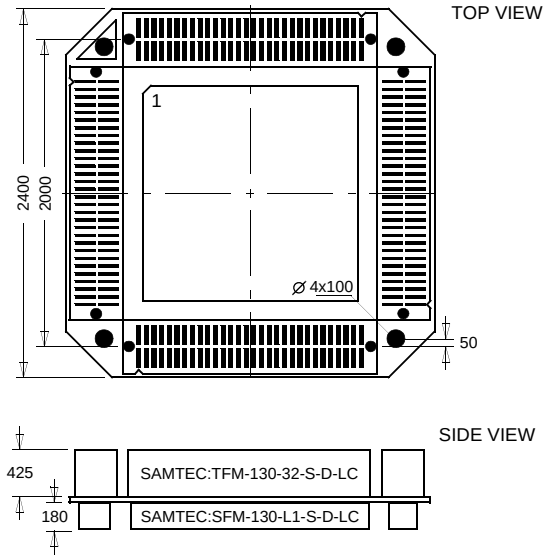
SIDE VIEW



TOP VIEW (all dimensions in mm)

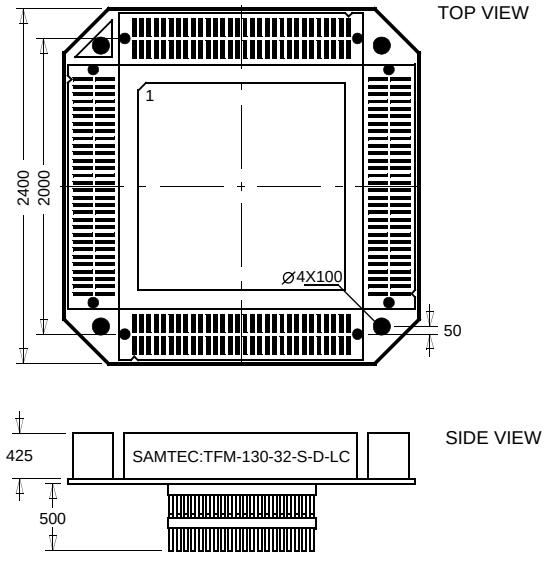
Dimension

LA-9539 BGA272-CPU-ADAPTER



ALL DIMENSIONS IN 1/1000 INCH

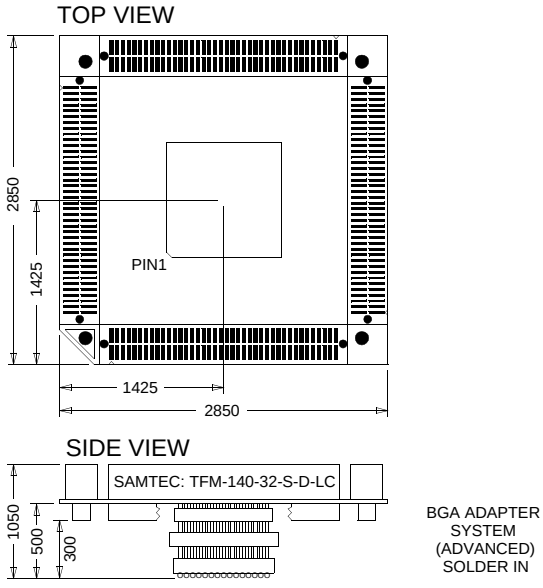
LA-9548 BGA272-AI-MPC555



ALL DIMENSIONS IN 1/1000 INCH

Dimension

LA-7907 TCON320-BGA357-PPC



ALL DIMENSIONS IN 1/1000 INCH

Not necessary.