SECTION 12
U-BUS TO IMB3 BUS INTERFACE (UIMB)

The U-bus to IMB3 bus interface (UIMB) Interface structure is used to connect the CPU internal unified bus (U-bus) to the intermodule bus 3 (IMB3). It controls bus communication between the U-bus and the IMB3.

The UIMB interface consists of seven submodules that control bus interface timing, address decode, data multiplexing, intrasystem communication (interrupts), and clock generation to allow communication between U-bus and the IMB3. The seven submodules are:

- U-bus interface
- IMB3 interface
- Address decoder
- Data multiplexer
- Interrupt synchronizer
- Clock control
- Scan control

12.1 Features

- Provides complete interfacing between the U-bus and the IMB3:
  - — 15 bits (32 Kbytes) of address decode on IMB3
  - — 32-bit data bus
  - — Read/write access to IMB3 module registers
  - — Interrupt synchronizer
  - — Monitoring of accesses to unimplemented addresses within UIMB interface address range
  - — Burst-inhibited accesses to the modules on IMB3
- Support of 32-bit and 16-bit BIUs for IMB3 modules
- Half and full speed operation of IMB3 bus with respect to U-bus
- Simple “slave only” U-bus interface implementation
  - — Transparent mode operation not supported
  - — Relinquish and retry not supported
- Supports scan control for modules on the IMB3 and on the U-bus

NOTE

Modules on the IMB3 bus can only be reset by $\overline{\text{SRESET}}$. Some modules may have a module reset, also.

WARNING

The user should not perform instruction fetches from modules on the IMB.

12.2 UIMB Block Diagram

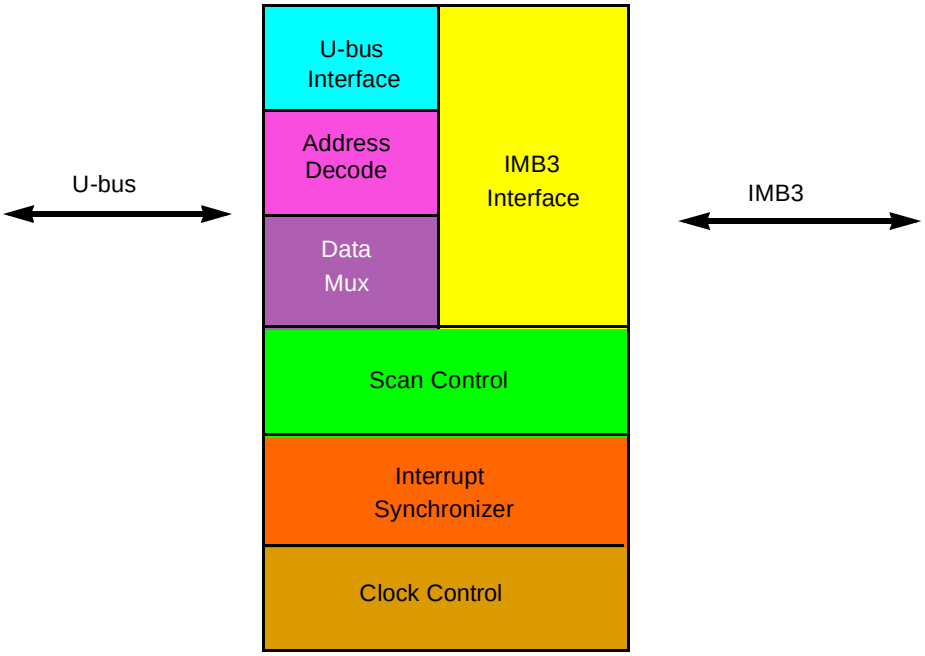

Figure 12-1 UIMB Interface Module Block Diagram

12.3 Clock Module

The clock module within the UIMB interface generates the IMB clock. The IMB clock is the main timing reference used within the IMB modules.

The IMB clock is generated based on the STOP and HSPEED bits in the UIMB module configuration register (UMCR). If the STOP bit is 1, the IMB clock is not generated. If the STOP bit is 0 and the HSPEED bit is 0, the IMB clock is generated as the inversion of the internal system clock. This is the same frequency as the CLKOUT if EBDF is 0b00 – full speed external bus. (See **Figure 12-2**.) If the HSPEED bit is 1, then the IMB clock is one-half of the internal system frequency. (See **Figure 12-3**.)

Table 12-1 STOP and HSPEED Bit Functionality

STOP	HSPEED	Functionality
0	0	IMB bus frequency is the same as U-bus frequency.
0	1	IMB bus frequency is half that of the U-bus frequency.
1	X	IMB clock is not generated.

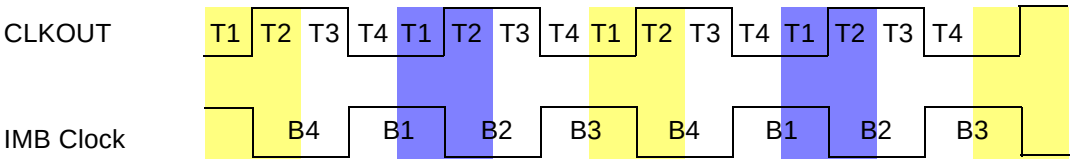

Figure 12-2 IMB Clock – Full-Speed IMB Bus

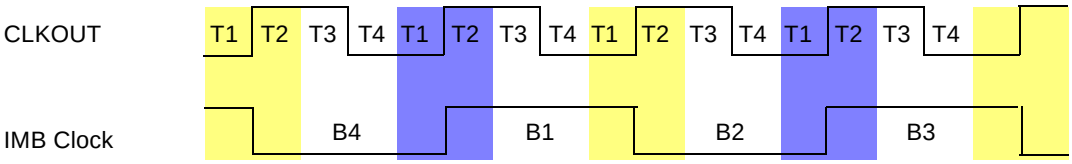

Figure 12-3 IMB Clock – Half-Speed IMB Bus

Table 12-2 shows the number of system clock cycles that the UIMB requires to perform each type of bus cycle. It is assumed in this table that the IMB3 is available to the UIMB at all times (fastest possible case).

Table 12-2 Bus Cycles and System Clock Cycles

Bus Cycle (from U-bus Transfer Start to U-bus Transfer Acknowledge)	Number of System Clock Cycles	
	Full Speed	Half Speed
Normal write	4	6
Normal read	4	6
Dynamically-sized write	6	10
Dynamically-sized read	6	10

NOTE

The UIMB interface dynamically interprets the port size of the addressed module during each bus cycle, allowing bus transfers to and from 16-bit and 32-bit IMB modules. During a bus transaction, the slave module on the IMB signals its port size (16- or 32-bit) via an internal port size signal.

12.4 Interrupt Operation

The interrupts from the modules on the IMB3 are propagated to the interrupt controller in the USIU through the UIMB interface. The UIMB interrupt synchronizer latches the Interrupts from the IMB3 and drives them onto the U-bus, where they are latched by the USIU interrupt controller.[1]

[1] The MPC565 / MPC566 includes an option for an enhanced interrupt controller. See **6.4.4 Enhanced Interrupt Controller** for operation details.

12.4.1 Interrupt Sources and Levels on IMB

The IMB3 has eight interrupt lines. There can be a maximum of 32 levels of interrupts from the modules on IMB bus. A single module can be a source for more than one interrupt. For example, the QSMCM can generate two interrupts (one for QSCI1/ QSCI2 and another for QSPI). In this case, the QSMCM has two interrupt sources. Each of these two sources can assert the interrupt on any of the 32 levels.

It is possible for multiple interrupt sources to assert the same interrupt level. To reduce the latency, it is a good practice for each interrupt source to assert an interrupt on a level on which no other interrupt source is mapped.

12.4.2 IMB Interrupt Multiplexing

The IMB has 10 lines for interrupt support. Eight lines are for interrupts and two for ILBS. These lines will transfer the 32 interrupt levels to the interrupt synchronizer. A diagram of the interrupt flow is shown in **Figure 12-4**.

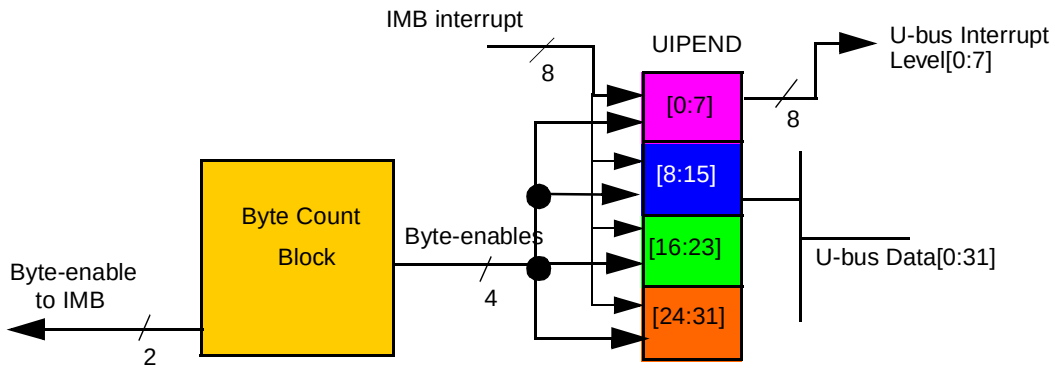

Figure 12-4 Interrupt Synchronizer Signal Flow

Latching 32 interrupt levels using eight IMB interrupt lines is accomplished with a 4:1 time-multiplexing scheme. The UIMB drives two signals (ILBS[0:1]) with a multiplexer select code that tells all interrupting modules on the IMB about which group of signals to drive during the next clock.

12.4.3 ILBS Sequencing

The IMB interface drives the ILBS signals continuously, incrementing through a code sequence (0b00, 0b01, 0b10, 0b11) once every clock The IRQMUX[0:1] bits in the IMB module configuration register select which type of multiplexing the Interrupt synchronizer will perform. The IRQMUX field can select time-multiplexing protocols for 8, 16, 24 or 32 interrupt sources. These protocols would take one, two, three or four clocks, respectively.

Table 12-4 shows ILBS sequencing. Programming IRQMUX[0:1] to 0b00 disables time multiplexing. In this case the ILBS lines remain at 0b00 at all times, as shown in **Table 12-4**. In this mode, no interrupts from IMB modules which assert on levels 8 through 31 are ever latched by the interrupt synchronizer. Time multiplexing is disabled during reset, but the reset default value enables time multiplexing as soon as reset is released if the reset default value is not 0b00.

The timing for the scheme and the values of ILBS and the interrupt levels driven onto the IMB IRQ lines are shown in **Figure 12-5**. This scheme causes a maximum latency of four clocks and an average latency of two clocks before the interrupt request can reach the interrupt synchronizer.

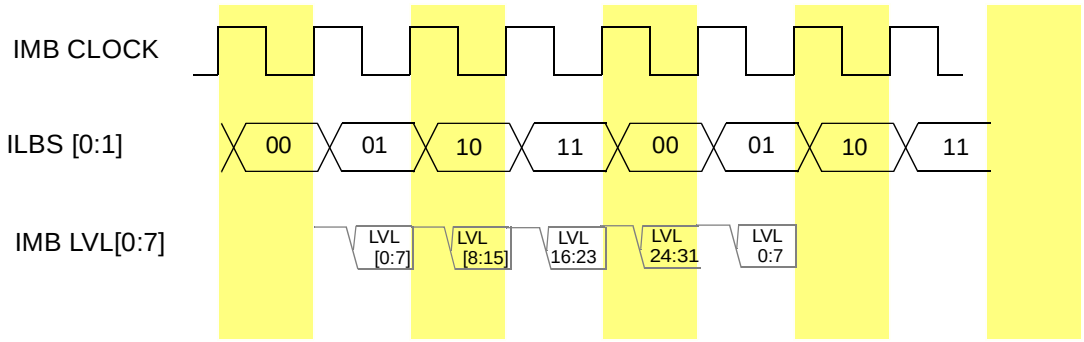

Figure 12-5 Time-Multiplexing Protocol for IRQ pins

Table 12-3 ILBS Signal functionality

ILBS[0:1]	Description
00	IMB interrupt sources mapped onto 0:7 levels will drive interrupts onto IMB IRQ[0:7]
01	IMB interrupt sources mapped onto 8:15 levels will drive interrupts onto IMB IRQ[0:7]
10	IMB interrupt sources mapped onto 16:23 levels will drive interrupts onto IMB IRQ[0:7]
11	IMB interrupt sources mapped onto 24:31 levels will drive interrupts onto IMB IRQ[0:7]

The IRQMUX bits determine how many levels of IMB interrupts are sampled. Refer to **Table 12-4**.

Table 12-4 IRQMUX Functionality

IRQMUX[0:1]	ILBS sequence	Description
00	00, 00, 00.....	Latch 0:7 IMB interrupt levels
01	00, 01, 00, 01....	Latch 0:15 IMB interrupt levels
10	00, 01, 10, 00, 01, 10,.....	Latch 0:23 IMB interrupt levels
11	00, 01, 10, 11, 00, 01, 10, 11,....	Latch 0:31 IMB interrupt levels

12.4.4 Interrupt Synchronizer

The interrupt synchronizer latches the 32 levels of interrupts from the IMB bus into a register which can be read by the CPU or other U-bus master. Since there are only eight lines for interrupts on the IMB and 32 levels of interrupts are possible, the 32 interrupt levels are multiplexed onto eight IMB interrupt lines. Apart from latching these interrupts in the register (UIPEND register), the interrupt synchronizer drives the interrupts onto the U-bus, where they are latched by the interrupt controller in the USIU.

If IMB modules drive interrupts on any of the 24 levels (levels eight through 31), they will be latched in the Interrupt pending register (UIPEND) in the UIMB. If any of the register bits 7 to 31 are set, then bit 7 will be set as well. Software must poll this register to find out which of the levels 7 to 31 are asserted.

The UIPEND register contains a status bit for each of the 32 interrupt levels. Each bit of the register is a read-only status bit, reflecting the current state of the corresponding interrupt signal. For each of the 32 interrupt levels, a corresponding bit of the UIPEND register is set.

Figure 12-4 shows how the eight interrupt lines are connected to the UIPEND register to represent 32 levels of interrupts. **Figure 12-6** shows the implementation of the interrupt synchronizer.

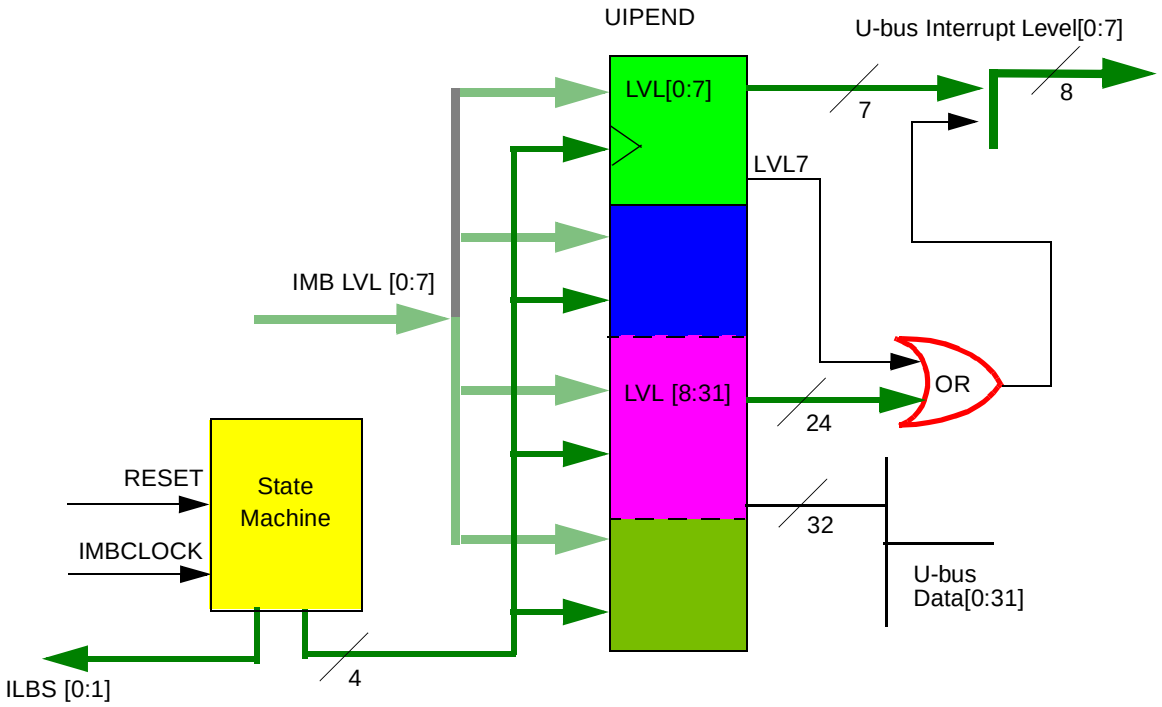

Figure 12-6 Interrupt Synchronizer Block Diagram

12.5 Programming Model

Table 12-5 lists the registers used for configuring and testing the UIMB module. The address offset shown in this table is from the start of the block reserved for UIMB registers. As shown in **Figure 1-2**, this block begins at offset 0x30 7F80 from the start of the MPC565 / MPC566 internal memory map (the last 128-byte sub-block of the UIMB interface memory map).

Table 12-5 UIMB Interface Register Map

Access	Base Address	Register
S[1]	0x30 7F80	UIMB Module Configuration Register (UMCR) See **Table 12-6** for bit descriptions.
—	0x30 7F84 — 0x30 7F8C	Reserved
S/T	0x30 7F90	UIMB Test Control Register (UTSTCREG) Reserved
—	0x30 7F94 — 0x30 7F9C	Reserved
S	0x30 7FA0	Interrupt Request Pending (UIPEND) See **12.5.3 Pending Interrupt Request Register (UIPEND)** for bit descriptions.

NOTES:
1. S = Supervisor mode only, T = Test mode only

Any word, half-word or byte access to a 32-bit location within the UIMB interface register decode block that is unimplemented (defined as reserved) causes the UIMB interface to asserting a data error exception on the U-bus.The entire 32-bit location must be defined as reserved in order for a data error exception to be asserted.

Unimplemented bits in a register return zero when read.

12.5.1 UIMB Module Configuration Register (UMCR)

The UIMB module configuration register (UMCR) is accessible in supervisor mode only.

UMCR — UIMB Module Configuration Register **0x30 7F80**

MSB

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
STOP	IRQMUX		HSPEED	RESERVED											
HRESET:															
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

LSB

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
RESERVED															
HRESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12-6 UMCR Bit Descriptions

Bit(s)	Name	Description
0	STOP	Stop enable. 0 = Enable system clock for IMB bus 1 = Disable IMB system clock To avoid complications at restart and data corruption, system software must stop each slave on the IMB before setting the STOP bit. Software must also ensure that all IMB interrupts have been serviced before setting this bit.
1:2	IRQMUX	Interrupt request multiplexing. These bits control the multiplexing of the 32 possible interrupt requests onto the eight IMB interrupt request lines. 00 = Disables the multiplexing scheme on the interrupt controller within this interface. What this means is that the IMB IRQ [0:7] signals are non-multiplexed, only providing 8 [0:7] interrupt request lines to the interrupt controller 01 = Enables the IMB IRQ control logic to perform a 2-to-1 multiplexing to allow transferring of 16 [0:15] interrupt sources 10 = Enables the IMB IRQ control logic to perform a 3-to-1 multiplexing to allow transferring of 24 [0:23]interrupt sources 11 = Enables the IMB IRQ control logic to perform a 4-to-1 multiplexing to allow transferring of 32 [0:31] interrupt sources
3	HSPEED	Half speed. The HSPEED bit controls the frequency at which the IMB3 runs with respect to the U-bus. This is a modify-once bit. Software can write the reset value of this bit any number of times. However, once logic 0 is written to this location, any attempt to rewrite this bit to a logic 1 will have no effect. 0 = IMB frequency is the same as that of the U-bus 1 = IMB frequency is one half that of the U-bus
4:31	—	Reserved

12.5.2 Test control register (UTSTCREG)

The UTSTCREG register is used for factory testing only.

12.5.3 Pending Interrupt Request Register (UIPEND)

The UIPEND register is a read-only status register which reflects the state of the 32 interrupt levels. The state of the IRQ[0] is shown in bit 0, the state of IRQ[1] is shown in bit 1 and so on. This register is accessible only in supervisor mode.

UIPEND — Pending Interrupt Request Register **0x30 7FA0**

MSB 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LVL0	LVL1	LVL2	LVL3	LVL4	LVL5	LVL6	LVL7	LVL8	LVL9	LVL0	LVL11	LVL12	LVL13	LVL14	LVL15
HRESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	LSB 31
LVL16	LVL17	LVL18	LVL19	LVL20	LVL21	LVL22	LVL23	LVL24	LVL25	LVL26	LVL27	LVL28	LVL29	LVL30	LVL31
HRESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12-7 UIPEND Bit Descriptions

Bit(s)	Name	Description
0:31	LVLx	Pending interrupt request level. Accessible only in supervisor mode. LVLx identifies the interrupt source as UIMB LVLx, where x is the interrupt number.