Appendix B7

General-Purpose I/O Port

GENERAL-PURPOSE I/O PORT

1. INTRODUCTION

When CL-GD546X is configured for the PCI bus, the BIOS ROM is accessed through CL-GD546X to avoid violating the ‘one load per adapter card’ PCI specification. The pins that access the ROM can implement an interface to one additional device on the adapter card. This is called GPIO (general-purpose I/O port). If the attached device does not respond in any way to bus activity when PCS# is high, GPIO accesses can be mixed with BIOS ROM accesses (usually, the ROM BIOS is only read at POST time).

Using the GPIO to access a second device on the adapter card has two major advantages over building a separate interface. First, it avoids violating the ‘one load’ specification. Second, it avoids the expense of additional decoding and control logic.

The GPIO is designed to connect to any one of three specific devices (see Section 2), but it can be used for any device whose host interface is compatible with one of the three target devices. It should be noted that device-specific software is required and will not necessarily be provided by Cirrus Logic.

The GPIO is available only when CL-GD546X is configured for the PCI bus.

2. CONFIGURATION

Three pins (two on the CL-GD5462) pins are optionally connected to pull-down resistors to configure the GPIO. These bits can be read back at MMI/O offset 1FEh. This is summarized in the Table B7-1.

Table B7-1. Local Peripheral Bus Configuration

V-Port™ Mode	GPIO 2:0 Configuration	
	CL-GD5462 RA[5:4]	CL-GD5464 RA[6:4]
C-CUBE CL480 (VMI mode ‘A’)	00	000
Reserved	01	001, 101–111
16-bit Intel® configuration	10	010
GPIO disabled	11	011
8-bit Intel® configuration (VMI mode ‘B’)	n/a	100

The GPIO Timing Control register is programmed to establish default timing for bus cycles. This is a 16-bit read/write register at MMI/O offset 1FCh.

This register contains four fields; each field establishes default timing for one phase of the GPIO cycle. Each field is programmed in terms of memory clock cycles.

Use the timing diagram in Figure B7-1 as an aid to the timing of the GPIO. There are five timing phases. The timing of the third phase is set by the local peripheral; the others are set by this register.

Table B7-2. GPIO Timing Control Register

Bits	Parameter	Description	Actual Clocks
15:12	t_1	Address, CS* hold from COMMAND	Equal to value programmed
11:8	t_2	RDY delay to COMMAND	Value programmed + 1
–	t_3	RDY delay (determined by peripheral device)	–
7:4	t_4	COMMAND delay to RDY	Value programmed + 1
3:0	t_5	Address, CS* setup to COMMAND	Value programmed + 1

The address and chip select are setup. These lines are ADDR in Table B7-3 on page B7-4. The number of address lines depends on how the GPIO is configured. After t_1, COMMAND is made active. The command lines are COMMAND in Table B7-3. Only one command is ever made active at any time. After t_2, the CL-GD546X begins to sample for a high on PRDY. If PRDY is low when the CL-GD546X begins sampling it, the operation delays until PRDY is high. This is shown in the diagram as t_3. The length of t_3 depends entirely on the peripheral device. If PRDY is high when it first sampled, there is a null t_3. Once PRDY is sampled high, CL-GD546X delays for t_4 and then makes COMMAND not active. On read cycles, the data from the peripheral is sampled one clock before the end of t_4. On write cycles, the data is valid with the address. After the command is made inactive, the CL-GD546X delays for t_5 and makes ADDR and CS* inactive. This completes the GPIO cycle.

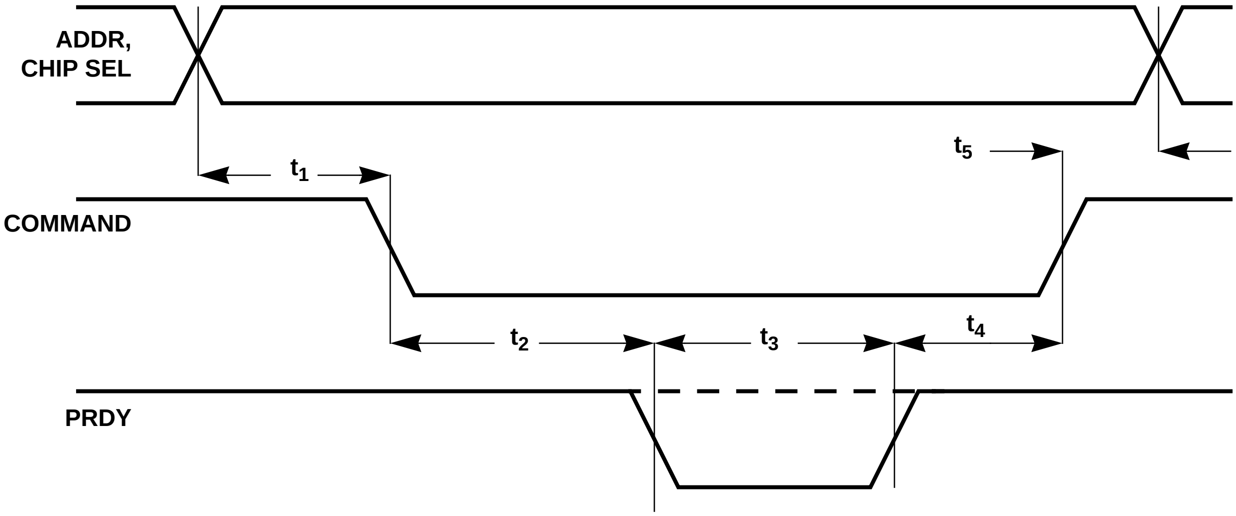

Figure B7-1. General-Purpose I/O Port Timing

3. PIN DEFINITIONS

Table B7-3 provides pin definitions for the EPROM and each of the three target devices. The pin names in the shaded column are the names given in the pin diagram and pin descriptions in the CL-GD546X data book.

Table B7-3. Pin Connections for General-Purpose I/O Port

Pin Number	Pin Name	EPROM	16-Bit Intel® Configuration	C-CUBE® CL480	8-Bit Intel® Configuration (CL-GD5464 Only)
117	RA[15]	A[15]	D[15]	HA[15]	n/c
115	RA[14]	A[14]	D[14]	HA[14]	n/c
113	RA[13]	A[13]	D[13]	HA[13]	n/c
112	RA[12]	A[12]	D[12]	HA12]	n/c
111	RA[11]	A[11]	D[11]	HA[11]	n/c
110	RA[10]	A[10]	D[10]	HA[10]	n/c
109	RA[9]	A[9]	D[9]	HA[9]	n/c
108	RA[8]	A[8]	D[8]	HA[8]	n/c
107	RA[7]	A[7]	IOW*	R/W*	IOR*
106	RA[6]	A[6]	IOR*	HA[6]	IOW*
104	RA[5]	A[5]	SA[5]	HA[5]	n/c
103	RA[4]	A[4]	SBHE*	HA[4]	n/c
102	RA[3]	A[3]	SA[3]	HA[3]	SA[3]
101	RA[2]	A[2]	SA[2]	HA[2]	SA[2]
100	RA[1]	A[1]	SA[1]	HA[1]	SA[1]
99	RA[0]	A[0]	SA[0]	HA[0]	SA[0]
97	RD[7]	D[7]	D[7]	HD[7]	D[7]
96	RD[6]	D[6]	D[6]	HD[6]	D[6]
95	RD[5]	D[5]	D[5]	HD[5]	D[5]
93	RD[4]	D[4]	D[4]	HD[4]	D[4]
92	RD[3]	D[3]	D[3]	HD[3]	D[3]
91	RD[2]	D[2]	D[2]	HD[2]	D[2]
90	RD[1]	D[1]	D[1]	HD[1]	D[1]
89	RD[0]	D[0]	D[0]	HD[0]	D[0]
88	PCS#	n/c	CS*	DS*	CS*
87	PRDY	n/c	IORDY	DTACK*	IORDY
62	ROMCS#	CE#/OE#	n/c	n/c	n/c

3.1 GPIO Data Register

The GPIO Data register is accessed with a group of addresses at MMI/O offset 180h through 1FFh. A write to any address in this range initiates one or more write cycles on the GPIO; a read to any address in this range initiates one or more read cycles from the GPIO. The least-significant bits of the actual address is the GPIO address.

The number and width (if appropriate) of the cycles depends on the byte lanes as shown in Table B7-4.

Table B7-4. GPIO Accesses

Byte Lanes	16-Bit Intel® Configuration	C-CUBE CL480	8-Bit Intel® Configuration (CL-GD5464 Only)
0000	Two 16-bit cycles	4 8-bit cycles	4 8-bit cycles
0001	One 16-bit cycle, One 8-bit cycle	Three 8-bit cycles	Three 8-bit cycles
0010	One 16-bit cycle, One 8-bit cycle	Three 8-bit cycles	Three 8-bit cycles
0011	One 16-bit cycle	Two 8-bit cycles	Two 8-bit cycles
0100	One 16-bit cycle, One 8-bit cycle	Three 8-bit cycles	Three 8-bit cycles
0101	Two 8-bit cycles	Two 8-bit cycles	Two 8-bit cycles
0110	Two 8-bit cycles	Two 8-bit cycles	Two 8-bit cycles
0111	One 8-bit cycle	One 8-bit cycle	One 8-bit cycle
1000	One 16-bit cycle, One 8-bit cycle	Three 8-bit cycles	Three 8-bit cycles
1001	Two 8-bit cycles	Two 8-bit cycles	Two 8-bit cycles
1010	Two 8-bit cycles	Two 8-bit cycles	Two 8-bit cycles
1011	One 8-bit cycle	One 8-bit cycle	One 8-bit cycle
1100	One 16-bit cycle	Two 8-bit cycles	Two 8-bit cycles
1101	One 8-bit cycle	One 8-bit cycle	One 8-bit cycle
1110	One 8-bit cycle	One 8-bit cycle	One 8-bit cycle
1111	No cycles	No cycles	No cycles