



Macrocad Development Inc. VCM 1286 Parallel Printer Port, IEEE1284 Synthesizable Host Model

Description:

Macrocad's VCM 1286 is a synthesizable IEEE1284 compliant host side RTL core model. It includes a 16 byte buffer for improved data communications. The VCM1286 model is designed for easy implementation and flexibility. The SHEL1286 model is the bi-directional buffer level. The CORE1286 contains the functional code required for the IEEE1284 printer port. The VCM1286 is configurable, and flexible. It supports DMA and interrupts. It supports automatic transfers for EPP, ECP and SPP modes.

Features:

- ? IEEE1284 spec compliant
- ? ECP, EPP 1.7, EPP 1.9 supported
- ? SPP, PS2, and FIFOed SPP modes supported
- ? DMA and FIFO operations supported
- ? 16 deep FIFO
- ? 1284 port time out features
- ? Flexible interrupts
- ? Digital signal filtering
- ? DMA and IRQ assignments programmable
- ? FIFO trigger level programmable
- ? Available in Verilog and VHDL versions
- ? Test Bench is included

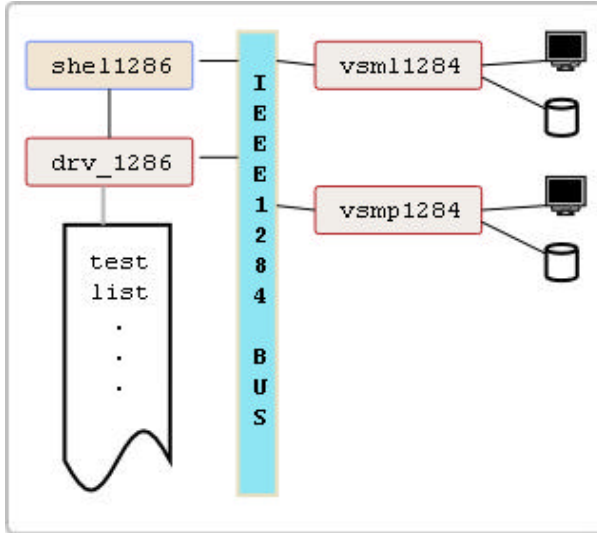


Figure 1



Architecture

The SHEL1286 model provides the bidirectional signals needed for the IEEE1285 peripheral. The CORE1285 contains the logic for the functions necessary to support EPP, ECP, SPP printer port modes. It contains the programming interface, register set, DMA and interrupt functions.

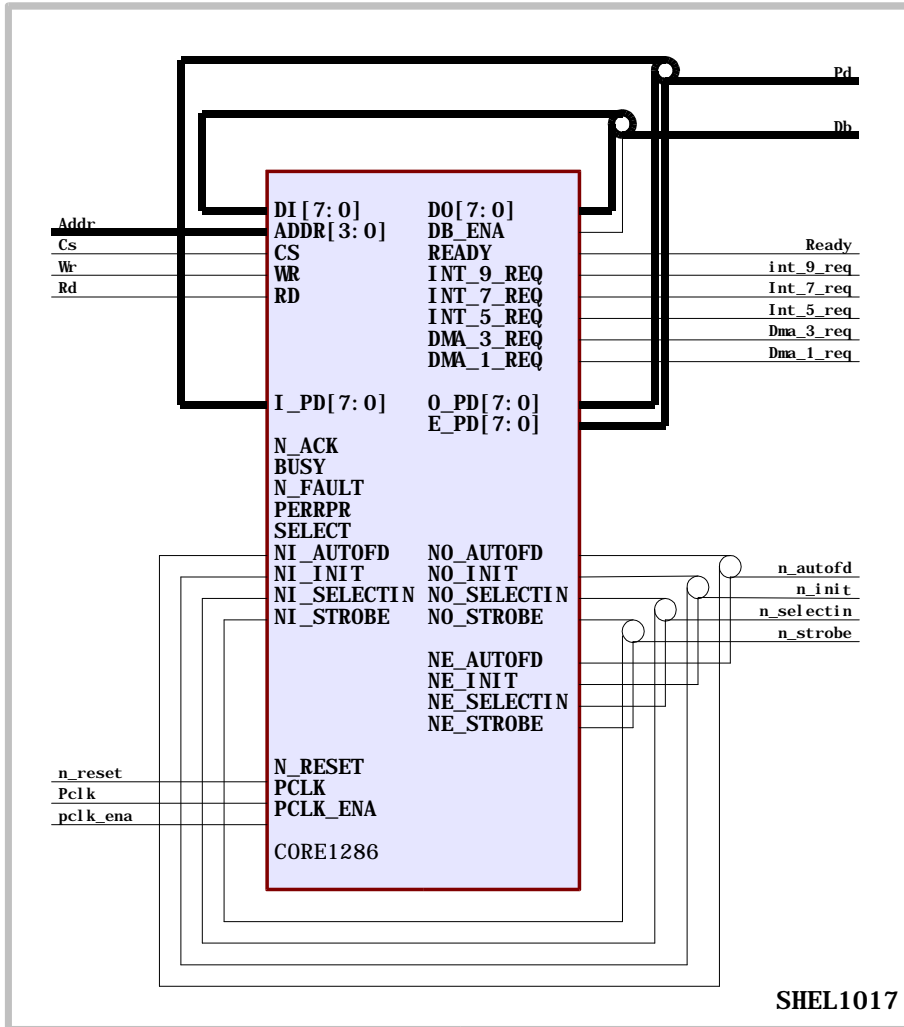
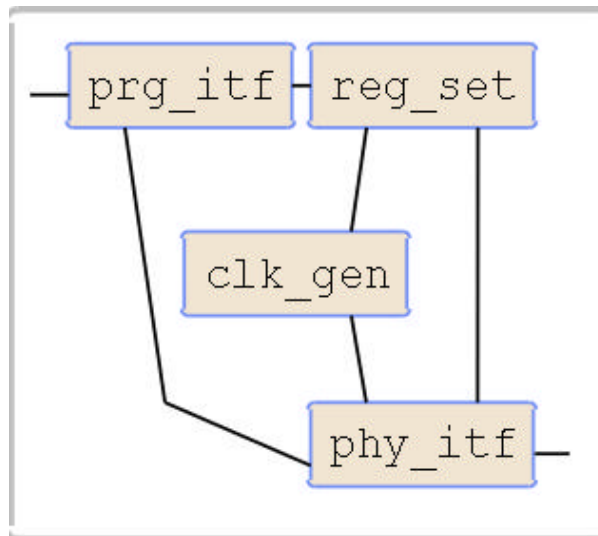


Figure 2



Functional Blocks

Access to the register set is accomplished via the “prg_itf” model. The “reg_set” model contains the registers which are considered status to IEEE1284 transfers. The “phy_itf” model accomplishes all the handshaking protocol necessary to transfer data on the IEEE1284 bus. The “clk_gen” acts as a prescaler for IEEE1284 timed operations.





IO programming access is shown below in figures 5 and 5.

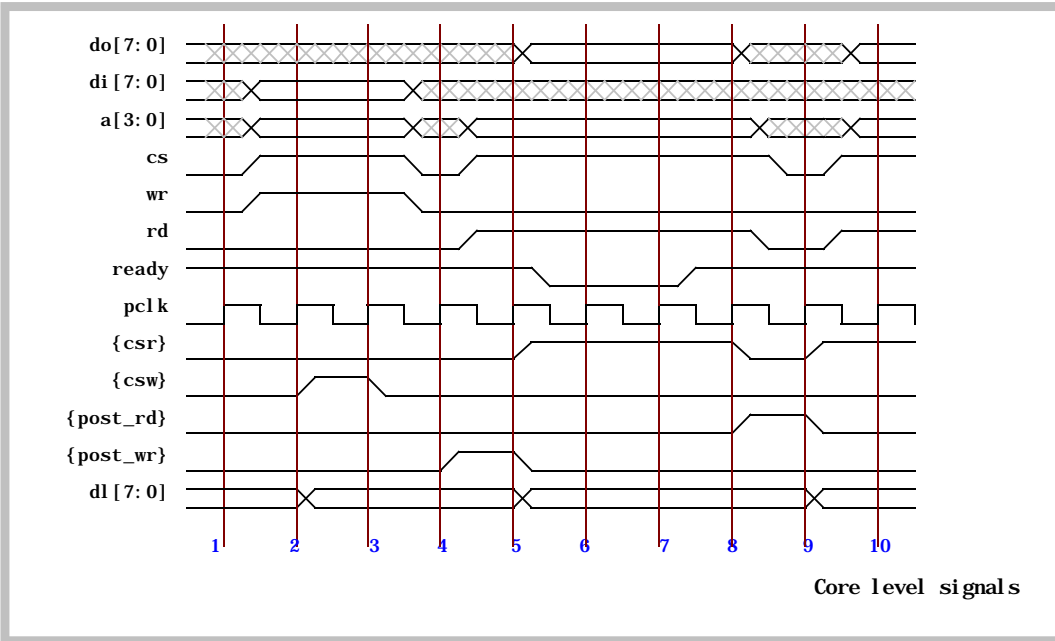


Figure 3 Condition 1 - not necessarily ready (EPP)

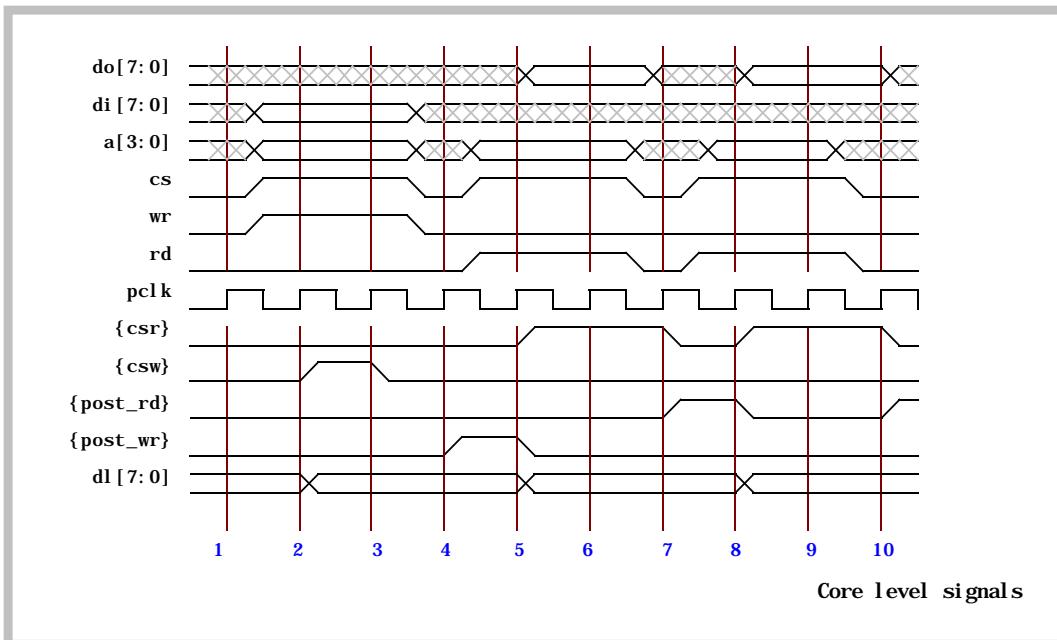


Figure 4 Condition 2 - guaranteed ready (NON-EPP)





Pins:

The VCM1286 shell contains the bi-directional signals for host implementations of the IEEE1284 port. The core level and below, contain only uni-directional signals.

Shell	Core	Type	Description
pd[7:0]	o_pd[7:0]	bi-directional	IEEE1284 Data bus
	i_pd[7:0]	out	IEEE1284 Data bus output
	e_pd[7:0]	in	IEEE1284 Data bus input
		out	Enable for IEEE1284 data bus
n_autofd	no_autofd	bi-directional	IEEE1284 host autofeed signal
	ne_autofd	out	Autofeed output
	ni_autofd	out	Autofeed enable
		in	Autofeed input
n_selectin	no_selectin	bi-directional	IEEE1284 host selectin signal
	ne_selectin	out	Selectin output
	ni_selectin	out	Selectin enable
		in	selectin input
n_init	no_init	bi-directional	IEEE1284 host init signal
	ne_init	out	init output
	ni_init	out	init enable
		in	init input
n_strobe	no_strobe	bi-directional	IEEE1284 host strobe signal
	ne_strobe	out	strobe output
	ni_strobe	out	strobe enable
		in	strobe input
n_ack	n_ack	in	IEEE1284 peripheral ack signal input
busy	busy	in	IEEE1284 peripheral busy signal input
n_fault	n_fault	in	IEEE1284 peripheral fault signal input
perror	perror	in	IEEE1284 peripheral perror signal input
select	select	in	IEEE1284 peripheral select signal input
db[7:0]	do[7:0]	bi-directional	Programming Data bus
	di[7:0]	out	Programming Data bus output
	db_ena	in	Programming Data bus input
		out	read enable for data bus
addr[3:0]	addr[3:0]	in	Programming register selects (address bus)
	cs	in	Chip select
	rd	in	Programming read enable
	wr	in	Programming write enable
ready	ready	out	Programming interface ready
int_9_req	int_9_req	out	Interrupt 9 request
int_7_req	int_7_req	out	Interrupt 7 request
int_5_req	int_5_req	out	Interrupt 5 request
dma_3_req	dma_3_req	out	DMA 3 request
dma_1_req	dma_1_req	out	DMA 1 request
n_reset	n_reset	in	Global reset
pclk_ena	pclk_ena	in	Global clock enable
pclk	pclk	in	Global clock

Table 1

**Register Set:**

Access to the register set, ("reg_set") is accomplished via the programming interface ("prg_itf") model. The registers are compliant with the Microsoft/HP definitions.

MODE	ADDRESS	OP	MODULE	NAME	DESCRIPTION
0	0	R/W	reg_set	pdo	1284 Data register
1	0	W	reg_set	pdo	1284 Data register
1	0	R	phy_itf	pdi	1284 Data pins
3	0	W	phy_itf	fad	ECP address / rle fifo
ALL	1	R	phy_itf	dsr	1284 Status register
ALL	2	R/W	reg_set	dcr	1284 Control register
4, 5	3	W	phy_itf	pad	EPP address register
4, 5	4	W	phy_itf	pd0	EPP data 0
4, 5	5	W	phy_itf	pd1	EPP data 1
4, 5	6	W	phy_itf	pd2	EPP data 2
4, 5	7	W	phy_itf	pd3	EPP data 3
2	8	W	phy_itf	fdo	SPP/ECP/TST data fifo
3	8	W	phy_itf	fdo	SPP/ECP/TST data fifo
3	8	R	phy_itf	fdi	SPP/ECP/TST data fifo
6	8	W	phy_itf	fdo	SPP/ECP/TST data fifo
6	8	R	phy_itf	fdi	SPP/ECP/TST data fifo
7	8	R	reg_set	cfa	Configuration A register
7	9	R/W	reg_set	cfb	Configuration B register
ALL	A	R/W	reg_set	ecr	Extended control register
ALL	B	R/W	reg_set	grn	Clock granularity
-	C	-	-	-	Reserved
-	D	-	-	-	Reserved
-	E	-	-	-	Reserved
-	F	-	-	-	Reserved

Table 2

The DCR, DSR, ECR, CFA, and CFB registers are for controlling operations of the VCM1284 port model. All other registers are data values to be transferred, (directly or indirectly) via the 1284 port.



Register Bit Mapping:

REGISTER	BIT	DESCRIPTION		
<i>cfa</i>	7	Configuration A register {read only = 0x14} {0} - irq are ISA compatible {pulsed}		
	6:4	{001} - Pword size is 8 bit		
	3	{0} - Reserved		
	2	{1} - no extra byte for fifo operations		
	1:0	{00} - na - sigle byte only		
<i>cfb</i>	7	Configuration B register {0} - compress disabled		
	6	irq signal state		
	5:3	irq select		
		111	: irq 5	
		110	: na	
		101	: na	
		100	: na	
		011	: na	
		010	: irq 9	
	2:0	dma channel selct		
		111	: na	
		110	: na	
		101	: na	
	<i>grn</i>	Clock granularity register		
		<i>ecr</i>	Extended control register	
7:5			Mode	
			111	: config mode
			110	: fifo test mode - diagnostic
	101		: epp 1.9 mode	
	100	: epp 1.7 mode		
	011	: ecp mode		
	010	: spp fifo mode		
	001	: ps2 mode		
000	: spp mode			
4	ecp interrupt mask bit			
3	ecp dma enable bit			
2	ecp service bit			
1	Fifo full bit			
0	Fifo empty bit			
<i>dcr</i>	Data control register			
	7:6	RESERVED		
	5	1284 data flow direction, {0 = forward, 1 = reverse}		
	4	Interrupt enable		
	3	1284 selectin		
	2	1284 init (inverted)		
	1	1284 autofeed		
0	1284 strobe			
<i>dsr</i>	data status register			
	7	1284 busy (inverted)		
	6	1284 ack (inverted)		
	5	1284 paper end		
	4	1284 Select		
	3	1284 fault (inverted)		
2:0	RESERVED			

Table 3

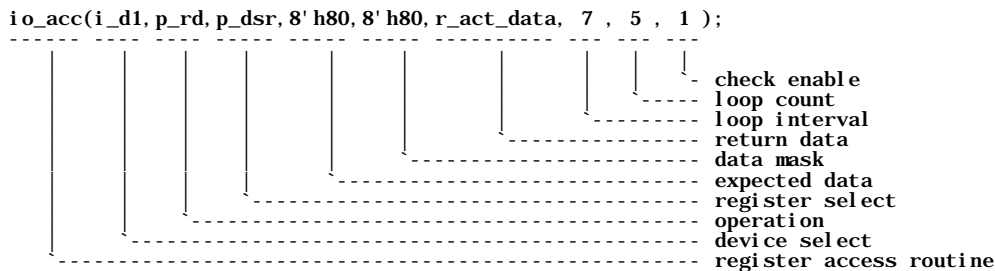
**Functional Description:**

The IEEE1284 host functions in the core1286 are divided into 4 models. The “prg_itf” provides a 3 clock access to the register set. There are two clock required for active read and write strobe, and a one clock recovery between any two register accesses. The “reg_set” contains the DCR, PDO, CFA, CFB, ECR and GRN registers. The “clk_gen” model provides the clock granularity counter to help pace the IEEE1284 transfers. The “phy_itf” model interfaces between the register set, and the IEEE1284 bus. It contains the state machines necessary to execute the IEEE1284 protocol for SPP, ECP and EPP transactions. Reverse byte and nibble modes are supported as well as reverse ID operations.



Test Bench:

The VCM1286 model comes with a test bench. The test bench consists of the sys_1286 model, the vsmp1284 peripheral, and the vsml1284 logger models. In figure 2 you can see how these models interact. Each model is independent of the others. The sys_1286 model instantiates the core, (shel1286), the core interface driver, (drv_1286), the bus logger and timing checker, (vsml1284), and the IEEE1284 peripheral, (vsmp1284) models. The sys_1286 model also contains the test list and test routines which determine the test content of the test bench operation. The test list is comprised of individual tests which concentrate on functions for the IEEE1284 bus. The format for the test instruction is shown below.



The “io_acc” routine allows for read and write access to any register in any device. In addition, the data can be masked on reads to ignore certain bits for the comapre feature. The driver will also loop on a register read access for the desired number of times with a specified interval between accesses. In the example above, the “io_acc” routine will perform a read access to the DSR register of the “i_d1” device. The expected data is 0x80, but bits 6:0 will be masked, and ignored. The routine will execute this read access to DSR 5 times, and have an interval of 7 us between io accesses. The data will be checked each time the io access is executed, but errors will not be reported until the last access of the loop. The “r_act_data” will contain the results of the last access in the loop.

**Reference Documents:**

DESCRIPTION	SOURCE	NAME
IEEE1284 specification	IEEE	1284-1994 IEEE Standard Signaling Method for a Bi-directional Parallel Peripheral Interface for Personal Computers (ANSI)
ISA implementation	Microsoft, Hewlett Packard	"The IEEE 1284 Extended Capabilities Port Protocol and ISA Interface Standard"

Table 4