Using the CompactFlash(CF) Interface as an External Data Bus

Applications

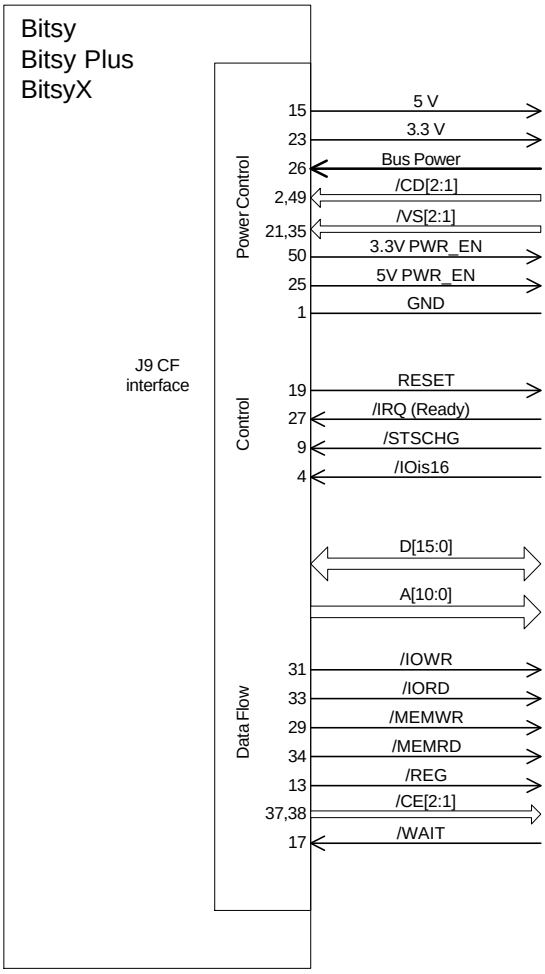
- High-volume data transfer (Ethernet, digital camera, multiple UARTs, parallel port, etc.)
- Supervisory and management (with low-latency response)
- Soft real-time control

Hardware

- CF bus has 11-bit, 3x2048kiB address space (PCMCIA uses a 26-bit, 64MiB address space)
- Three addressable regions: I/O, Memory and Attribute
- Bits in StrongARM MECR register set access timing for each region

Bus Signals

Bus Power (CardBVcc)	Voltage used to power bus signals.
PWR_EN	Power Enable GPIOs
/CD[2:1]	Card Detects. To power the bus signals, detect pins (/CDn) must be pulled low.
/MEMRD, /MEMWR	Memory read/write or Attribute space read
/REG	0=Attribute space, 1=Memory space
/CE[2:1]	Indicate the memory space data transfer type (8-, 16-bit data)
/IORD, WR	I/O space read/write
/IOis16	0=I/O data is 16-bit, 1=8-bit
/IRQ	Interrupt Request to processor
/WAIT	CPU waits for this signal to go high before completing a CF bus operation. Use carefully, as ALL CPU activity stops while this line is low. PCCR register can disable this signal.
/STSCHG	Status Change interrupt
/RESET	GPIO to reset device, if needed



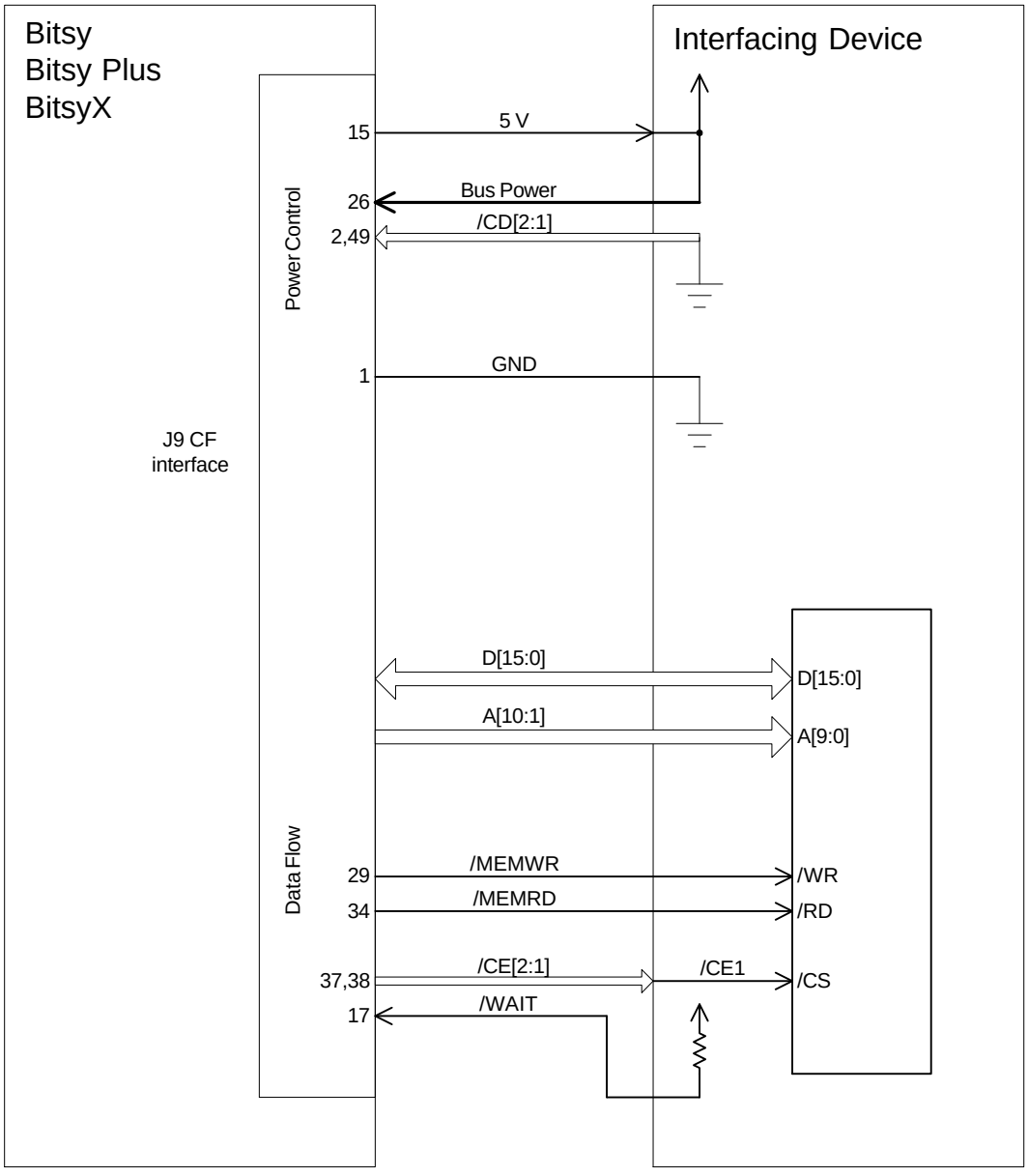
Address Space	CF: "Socket 1"	PCMCIA: "Socket 0"
I/O space	0x3000 0000 to 3000 07ff	0x2000 0000 to 23ff ffff
Attribute space	0x3800 0000 to 3800 07ff	0x2800 0000 to 2bff ffff
Memory space	0x3C00 0000 to 3c00 07ff	0x2C00 0000 to 2fff ffff

Sample Application

The following example maps a 5V, 16-bit device to the memory space of the CF bus. Unused bus signals have been removed for clarity.

Notes:

- The CPU sets /CE[2:1] to logic 1 and 0, respectively, during 16-bit access. This design assumes that software always performs 16-bit writes. Decoding logic can be added, if desired.
- CF bus A1 maps to device's A0, since CF addresses are for 8-bit data.
- This design is not set up for power management. The device in this example will draw power from the ADS system even when the system is in Sleep mode



O/S, General Issues

- Disable compact flash driver to avoid confusion or resource contention
- Screen may flicker if device uses long and/or slow memory cycles

Windows CE

- Requires Platform Builder to create device driver, as there are a lot of header and LIB files needed that aren't included with SDK or even BSP.
- For applications that can tolerate latency, or are polled, could create a driver that maps to CF. System issues a Windows event when an interrupt occurs.
- DDTK includes an interrupt latency tester (uses system timer)
- CAN driver could be used as model for how to create and load an external device driver

Linux

- Map device driver to CF region of memory
- Map interrupt handler to S1ReadynInt
- Disable CF driver

References

- Bitsy Plus Users Manual. Applied Data Systems, 2002.
- Fiasal. Inside PCCard: CardBus and PCMCIA Design (EDN Series for Design Engineers), Butterworth-Heineman. 1996
- Intel. SA1110 Reference Manual, *Section 10.7: PCMCIA*. (Order Number: 278240-004). October 2001. pp. 169-177
- Intel. SA1111 Developer's Manual,*Chapter 12: PCMCIA* (Intel Order No: 278242-003). July 2000.
- Mori and Welder. PCMCIA Developers Guide

Document Revision History

REV	DESCRIPTION	DATE	BY
1	Preliminary document	9/18/2001	ak
2	Initial release Add signal descriptions, updated block diagram, references and application example	3/14/2003	ak
3	Correct /IOis16 signal polarity and values Add Socket 0/PCMCIA addresses List full address range of CF and PCMCIA Add this revision history	3/18/2003	ak
	Add note about Wait signal Indicate that StsChg can be used as an interrupt	4/8/2003	ak