RCBus Specification

Pre-release
for review and
feedback

Version 0.0.003
27-February-2023

Introduction

The documentation describes the Retro Computer Bus (RCBus).

The RCBus is an extended version of the RC2014™ bus[1].

The RCBus has been created for the following reasons:
1. To help distinguish RC2014™ products from products that are not RC2014™ but that claim a degree of compatibility with RC2014™
2. To reduce the chances of trademark infringement
3. To help members of the retro computer community share designs and sell products based on the RC2014™ bus
4. To specify a common approach to supporting features not provided by the RC2014™ bus
5. To specify a common approach to supporting processors other than the Z80

Appendix A provides some background information which should help explain the above list of reasons for creating the RCBus.

The RCBus has a common backplane but some of the signals are not required for some use cases and some signals have different functions depending on the processor and other factors. Modules connected to the backplane can either work with all other RCBus modules or, more likely, a sub-set of RCBus modules.

To indicate which modules are compatible with each other, a number of sub-specifications are documented, such as:

1.	RCBus-2014	This is the RC2014™ specification (see www.rc2014.co.uk)
a)	RCBus-2014-s	RC2014™ standard (40-pin) bus
b)	RCBus-2014-e	RC2014™ enhanced (>40-pin) bus
2.	RCBus-Z80	Includes extensions to support advanced Z80 and Z180 family features
3.	RCBus-68xx	Includes extensions to support the Motorola bus style processors

A module that only requires the RC2014™ standard 40-pin bus might be compatible with RC2014-2014-s, RCBus-2014-e, and RCBus-Z80. Of course, such a compatibility claim does not mean you can put any two “compatible” modules together and expect them to work. For example, two RC2014™ serial modules will not work due to address conflicts. What “compatible” means is the physical and electrical bus signals are compatible.

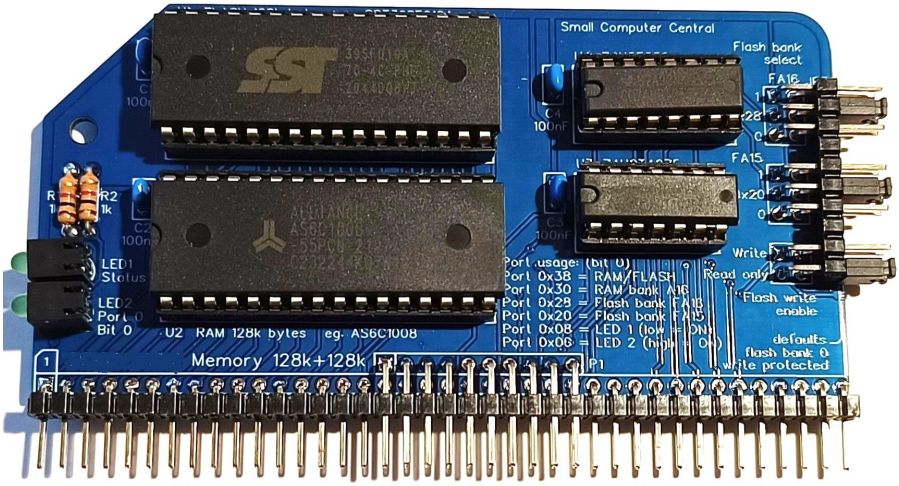
The term RCBus can be used for any design that works with the full RCBus backplane or a sub-set of the RCBus backplane.

[1] RC2014 is a trademark under British law, belonging to RFC2795 Ltd (Spencer Owen’s company)

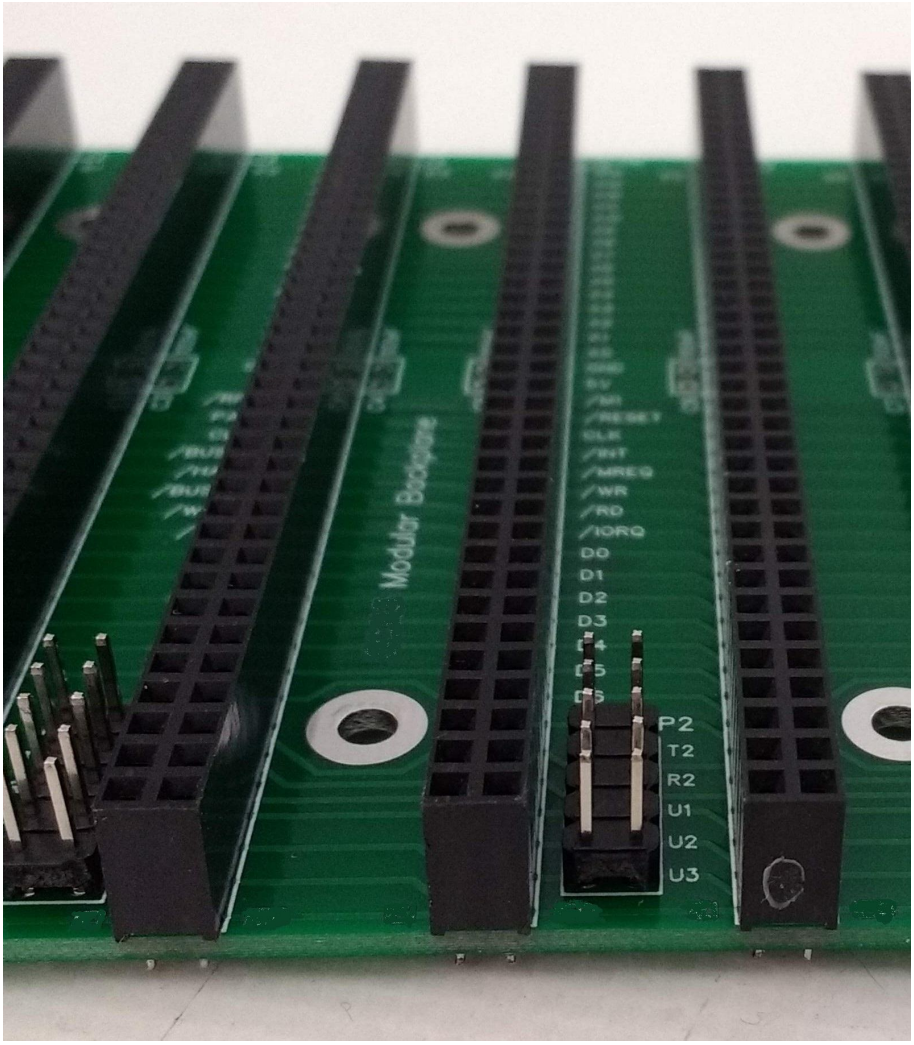
Backplanes and Modules

Backplanes and modules can be any size and shape, but established norms are worth following.

Below is an example of an RCBus module which has a partial second row of bus pins.

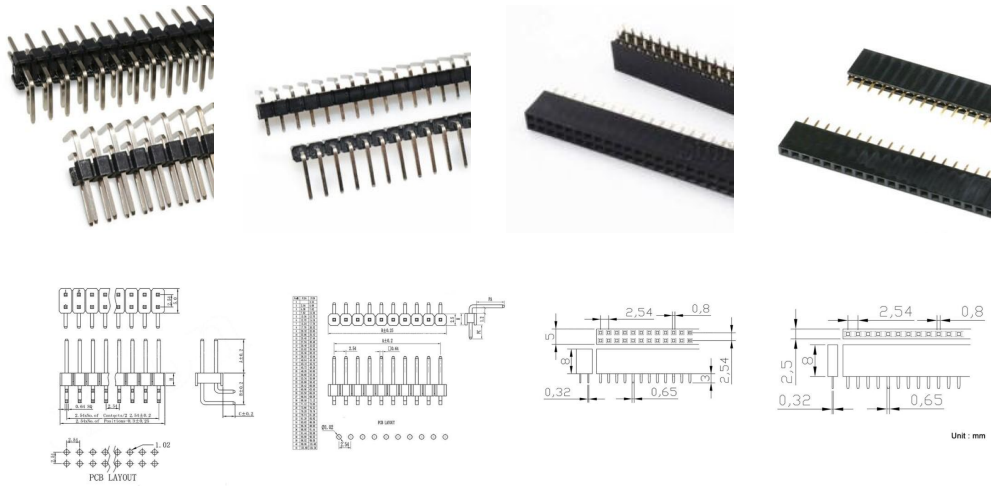


Below is an example of an RCBus backplane which has 80-pin module sockets.



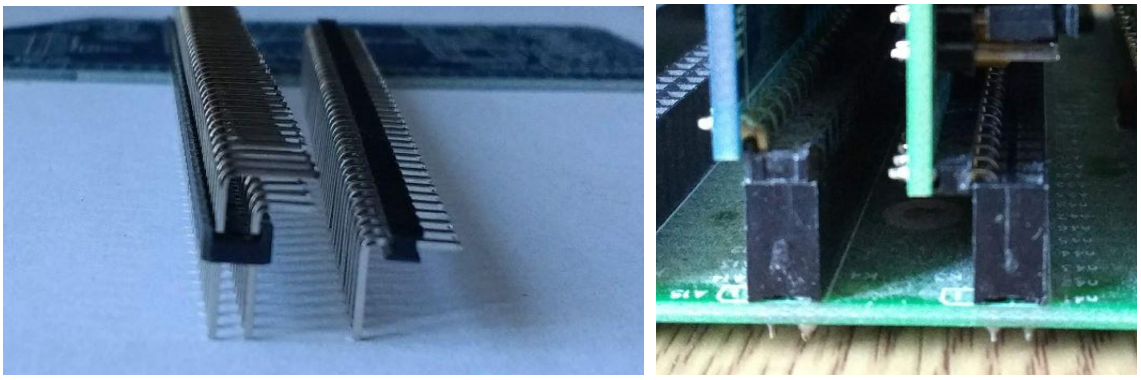
Bus Connectors

The bus connectors are low cost header pins and sockets, either single or double row with up to 40-pins in each row.



Pin length ~6mm, pitch 2.54mm

Note that the most commonly used male 1x40 angled connector has a different profile to the 2x40 connector.



Single row bus socket viewed from above:

Pin 1 Pin 40

Front *Back*

Double row bus socket viewed from above:

Pin 1 Pin 40

Pin 41 Pin 80

Front *Back*

Bus Pin Assignments

Bus signals are either common to all bus specifications or specific to one or more specifications.

Pin 1st row	**RC2014™ standard**	**RCBus backplane**	**RCBus Z80**	**RCBus 68xx**	**RCBus 9995**	**Notional 16-bit CPU**		
1	A15							
2	A14							
3	A13							
4	A12							
5	A11							
6	A10							
7	A9							
8	A8							
9	A7							
10	A6							
11	A5							
12	A4							
13	A3							
14	A2							
15	A1							
16	A0							
17	GND							
18	+5V							
19	M1							
20	RESET							
21	CLOCK							
22	INT							
23	MREQ							
24	WR							
25	RD							
26	IORQ							
27	D0							
28	D1							
29	D2							
30	D3							
31	D4							
32	D5							
33	D6							
34	D7							
35	TX							
36	RX							
	2014	*Backplane*	*Z80*	*68xx*	*9995*	*16-bit*		
37	USER1	n37	INT1	FIRQ	MEMEN	n37		
38	USER2	n38 [1]	IEI	E	CRUIN	n38 [1]		
39	USER3	n39 [1]	IEO	RW	CRUCLK	n39 [1]		
40	USER4	n40	n40	n40	n40	n40		

[1] Backplane configurable for direct or cascade (daisy-chain) connections

Second row bus signals:

Pin 2nd row	RC2014™ enhanced	RCBus Backplane	RCBus Z80	RCBus 68xx	RCBus 9995	Notional 16-bit CPU		
41	N/A	3v3						
42	N/A	n42 [1]	BAI	n42	n42	n42		
43	N/A	n43 [1]	BAO	n43	n43	n43		
44	N/A	n44 ? USER#	n44	n44	n44	n44		
45	N/A	n45 ? HiLo byte	n45	n45	n45	n45		
46	N/A	n46	n46	n46	n46	n46		
47	N/A	n47	n47	n47	n47	n47		
48	N/A	n48	n48	n48	n48	n48		
49	N/A	A23						
50	N/A	A22						
51	N/A	A21						
52	N/A	A20						
53	N/A	A19						
54	N/A	A18						
55	N/A	A17						
56	N/A	A16						
57	GND							
58	+5V							
59	RFSH							
60	PAGE							
61	CLOCK2							
62	BUSACK							
63	HALT							
64	BUSRQ							
65	WAIT							
66	NMI							
67	D8							
68	D9							
69	D10							
70	D11							
71	D12							
72	D13							
73	D14							
74	D15							
75	TX2							
76	RX2							
	2014	*Backplane*	*Z80*	*68xx*	*9995*	*16-bit*		
77	USER5	n77	INT2	n77	n77	n77		
78	USER6	n78	n78	n78	n78	n78		
79	USER7	n79	n79	n79	n79	n79		
80	USER8	n80	n80	n80	n80	n80		

[1] Backplane configurable for direct or cascade (daisy-chain) connections

Bus Signals

Signals are TTL level unless otherwise indicated.

Common Signals

This section describes each of the signals common to all processor types.

Signals are defined by their function with a Z80 CPU unless otherwise stated.

+5V	Power supply +5 volts, recommended tolerance +/-0.25 volts.
+3V3	Power supply +3.3 volts. Optional.
A0 to A23	Address bus, output from CPU, active high, tristate. Typically, 8-bit processors will only use A0 to A15.
BUSAK	Bus acknowledge, output from CPU, active low. The CPU outputs a low on this line indicating the address bus, data bus, and control signals MREQ, IORQ, RD, and WR have entered their high-impedance states.
BUSRQ	Bus request, input to CPU, active low. This is pulled low by a device that wants to take control on the CPU's address, data and control bus.
CLOCK	Clock, input to CPU. System clock used for synchronisation so it should normally be the CPU clock signal.
CLOCK2	This is a second clock usually used as a clock source for a UART.
D0 to D15	Data bus, input/output, active high, tristate. 8-bit processors will only used D0 to D7. Unused data lines can be left floating.
GND	Power supply ground.
HALT	Halt, output from CPU, active low. This indicates the CPU has executed a HALT instruction and is waiting for an interrupt before resuming operation.
INT	Interrupt request, input to CPU, active low. This signal should have a pull-up resistor can is pulled low by any device requesting an interrupt.
IORQ	Input/output request, output from CPU, active low, tristate. The Z80 has a separate address space for I/O devices but other processors will likely need to create a window in their memory map for I/O.
M1	Machine cycle one, output from CPU, active low. This signal is very specific to the Z80 and should be pulled up for other processors.
MREQ	Memory request, output from CPU, active low, tristate. Indicates that the address bus holds a valid address for a memory read or a memory write operation.
NMI	Non-maskable interrupt, input to CPU, active low. Negative edge triggered interrupt.
PAGE	Page RAM/ROM, active high. Low to enable ROM and disable its shadow RAM. High to disable ROM and enable its shadow RAM. For a Z80 system the ROM is at the bottom of memory (starting at address 0x0000) following reset. This signal is set high to replace this

ROM with RAM, allowing a full 64k of RAM. Perhaps a better name would be RAM_ENable. There are other memory management schemes in use that do not use this signal.

RD — Read, output from CPU, active low, tristate. Indicates that the CPU wants to read data from memory or an I/O device.

RESET — Reset, input to CPU, active low. System reset signal.

RFSH — Refresh, output from CPU, active low. Indicates a memory refresh cycle.

RX — This signal is designed to allow a serial port to communicate with a device on another module. RX is an input to a serial port, such as a UART, and an output from a serial device, such as a terminal module.

RX2 — A second serial communications 'RX' signal.

TX — This signal is designed to allow a serial port to communicate with a device on another module. TX is an output from a serial port, such as a UART, and an input to a serial device, such as a terminal module.

TX2 — A second serial communications 'TX' signal.

WAIT — Wait, input to CPU, active low. Communicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a WAIT state as long as this signal is active.

WR — Write, output from CPU, active low, tristate. Indicates that the CPU wants to write data to memory or an I/O device.

RCBus-2014

The following are signals specific to the basic RC2014™ bus specification.

USER# — Signals USER1 to USER8 are free for the user to do as they please.

RCBus-Z80

The following are signals specific to systems using the Z80 family bus specification. Eg. Z80 and Z180.

BAI — Bus acknowledge in, input, active low. BAI and BAO form a direct memory access priority daisy-chain. BAI signals that the system buses have been released for DMA control.

BAO — Bus acknowledge out, output, active low. BAI and BAO form a direct memory access priority daisy-chain. BAO signals that the CPU has relinquished control of the bus.

IEI — Interrupt enable in, input to an interrupt generating device, active high. IEI and IEO form an interrupt priority daisy-chain. A high on IEI indicates that no other device of higher priority is interrupting.

IEO — Interrupt enable out, output from an interrupt generating device, active high. IEI and IEO form an interrupt priority daisy-chain. IEO is high only when IEI is high and this device is not requesting an interrupt.

INT1 — Interrupt, input to CPU, active low.

INT2 — Interrupt, input to CPU, active low.

n## Reserved

RCBus-68xx

The following are signals specific to systems using the 68xx, 63xx and 65xx family bus specification.

FIRQ Fast interrupt request, input to CPU, active low.

E Data bus enable, input to CPU, active high.

RW Read/write, output from CPU. High for read operations, low for write operations.

n## Reserved

RCBus-9995

The following are signals specific to systems using the TMS9995 family bus specification.

CRUCLK CRU clock, output from CPU, active low.

CRUIN CRU input data,input to CPU, active high.

MEMEN Memory enable, output from CPU, active low.

n## Reserved

Guidance for Designers

The basic idea is that all processor types generate the common bus signals, essentially Z80 compatible signals, where ever possible. Any additional processor or sub-specification specific signals are optional. This approach is designed to give the greatest compatibility across all uses of the RCBus.

It is recommended that CMOS ICs be used wherever possible and that 74HCT series ICs are used rather than other 74 series families.

For further guidance on processor specific issues and sub-specification specific issues, see the appropriate appendix.

Sub-specification specific signals

It is recommended that jumper options be provided on modules for connections to the sub-specification specific signals. Namely, pins 37 to 40, 77 to 80, and 42 to 48.
??? should this only be the traditional USER pins? (ie. 37 to 40, and 77 to 80)

Address bus

Unused address lines should ideally be pulled down with weak pulled down resistors to enable CPU and memory modules to be used together even if there is a mismatch in the number of address lines used.

Memory map

Some processors require boot ROMs to be at the top of memory and others at the bottom of memory. It is recommended that memory modules provide an option for this.

Some processors have separate address spaces for I/O devices and some require them to be mapped as memory. Memory modules designed to work with different processors will need to allow for this.

Processors that do not have separate I/O address spaces should generate the RCBus IOREQ signal when I/O addresses are decoded and also ensure the MREQ signal is not active at these times.

Unresolved Issues & Notes

Possible additional signals needed: (Alan Cox)
No low/high enables to make D15-D8 actually usable
Multiple interrupt lines
DMA - BUSRQ etc
3v3 power

Are TX2 and RX2 required? (Tadeusz Pycio)

Do we need DREQ and TEND for Z180

Possibly 3v3 supply (but as Tadeusz has pointed out existing backplanes only have thin tracks on the available signal lines)

16-bit data hi/lo byte enable (1 pin or 2)

Some new pins as genuine USER functions‼

Do we want a MEM_EN signal? (Tadeusz Pycio)

Better name for PAGE = RAM_EN ? (Tadeusz Pycio)

We still do not have assigned pins for the DMA service request signals. My suggestion was to use pins 78 (DREQ1) and 79 (DREQ2), but if others see the need to add DACK/EOP in addition to these signals it is worth considering another location (pins 45-48 ?) (Tadeusz Pycio)

Appendix A - Background

To understand what the RCBus is and why it exists, it is necessary to consider what has led us here.

In 2014 Spencer Owen created a modular version of Grant Searle's Z80 computer design, which he called RC2014. He began selling it on Tindie and in 2016 he gave up his job and worked on RC2014 full time.

The RC2014 system attracted a community of enthusiasts, some of which made RC2014 compatible modules, as encouraged by Spencer's website: "If your module may be of use to other RC2014 owners, please consider sharing your design or selling them yourself. I'm happy to help you with this and to spread the word. Note that "RC2014" is a registered trademark, so you are not allowed to call your module "RC2014 [thingy] Module" or use the RC2014 logo. However, feel free to mark your modules as "Designed for RC2014."

Before long there was talk of extensions to the RC2014 bus.

In 2018 the topic "New backplane -wishes ?" led to Spencer posting the following on 17-June-2018:

"I've been musing over enhancements to the backplane for a little while now, and whilst nothing is set in stone, the pin layout would follow this;

Enhanced		Standard			Enhanced		Standard	
A31	1	1	A15		Clock2	21	21	Clock
A30	2	2	A14		BUSACK	22	22	INT
A29	3	3	A13		HALT	23	23	MREQ
A28	4	4	A12		BUSRQ	24	24	WR
A27	5	5	A11		WAIT	25	25	RD
A26	6	6	A10		NMI	26	26	IORQ
A25	7	7	A9		D8	27	27	D0
A24	8	8	A8		D9	28	28	D1
A23	9	9	A7		D10	29	29	D2
A22	10	10	A6		D11	30	30	D3
A21	11	11	A5		D12	31	31	D4
A20	12	12	A4		D13	32	32	D5
A19	13	13	A3		D14	33	33	D6
A18	14	14	A2		D15	34	34	D7
A17	15	15	A1		Tx2	35	35	Tx
A16	16	16	A0		Rx2	36	36	Rx
Gnd	17	17	Gnd		USR5	37	37	USR1
5v	18	18	5v		I2C SDA	38	38	IEI
RFSH	19	19	M1		I2C SCL	39	39	IEO
Page	20	20	Reset		USR8	40	40	USR4

However, on 16-June-2019 Spencer created the topic "Upcoming changes to the RC2014 bus and ecosystem" in which he stated "RC2014 will not be changing" and further clarified this by stating "The RC2014 bus does not support IEI/IEO modules. Through-hole components are used. And the physical bus will not be changing."

This came as a bit of a blow to those who were looking to build on the RC2014 system and led to a discussion about how to move forward. This discussion didn't produce any solid answers.

On 31-Jan-2023 Spencer created the topic “What has an RC2014 and a Hoover got in common?” in which he stated the following:
”RC2014 is a trademark under British law, belonging to RFC2795 Ltd (ie my company).”
“All of these kits carry the RC2014 name and RC2014 logo, and are labelled as being RFC2795 Compliant.”
“Any other kit is NOT an RC2014.”
“All sellers seem to do a very good job of making the distinction in their listings. But I don't think it is doing anybody any favours by calling a non-RC2014 machine an RC2014, least of all to the creators of compatible machines.“

This led to another discussion about the future in which it was suggested that a new name be found for products that have a degree of compatibility with RC2014 products but are not made by RFC2795 and are thus not RC2014 products. To this suggestion, Spencer wrote: “RCBus or RC80 Bus sound good to me. It takes the essence of what the bus is without limiting it by what the RC2014 natively supports.”

Spencer’s official description of an RC2014 remains:
“RC2014 is a simple 8 bit Z80 based modular computer. It is inspired by the home built computers of the late 70s and computer revolution of the early 80s. It is not a clone of anything specific, but there are suggestions of the ZX81, UK101, S100, Superboard II and Apple I in here. It nominally has 8K ROM, 32K RAM, runs at 7.3728MHz and communicates over serial at 115,200 baud.”

Much of what some in the retro computer community wish to do with their RC2014 based systems does not match this description.

And thus the RCBus project was created.

Appendix B - RC2014™ USER Pin Usage

The RC2014™ bus has a number of spare pins, usually called USER pins. These have been used by designers to add functions not provided by the defined bus pins. The RCBus specification attempts to maintain compatibility with the most common uses. The following is a list of some of those uses.

Product	Pin 37	Pin 38	Pin 39	Pin 40	Pin 77	Pin 78	Pin 79	Pin 80
RC2014	USER1	USER2	USER3	USER4	USER5	USER6	USER7	USER8
BP80	USER1	USER2	USER3	IEO	USER5	USER6	USER7	IEI
SC102 Z80 CTC	BCT3*	IEI*	IEO*		BCT0*	BCT1*	BCT2*	
SC103 Z80 PIO		IEI*	IEO*					
SC104 Z80 SIO/2		IEI*	IEO*					
SC110 SIO+CTC	CTC3*	IEI*	IEO*		CTC0*			
SC111 Z180 CPU	INT1*							
SC112 Backplane				IEO				IEI
SC113 Backplane				IEO				IEI
SC116 Backplane				IEO				IEI
SC126 Z180 SBC				IEO		SCL* (I2C)	SDA* (I2C)	IEI
SC132 Z80 SIO/0		IEI*	IEO*					
SC149 Z80 CPU	BUSAK*	WAIT*	BUSAK*	NMI* BUSAK*				
Z80 CPU + CTC module (TP)	CTC3*	IEI*	IEO*		CTC0*			
Z180 MPU (TP)	INT1*				INT2*			
Z280 MPU (TP)	INTA*				INTC*			
Universal SIO (TP)		IEI*	IEO*					
DUART (TP)		IEI*	IEO*					
16450/550 (TP)	IRQ*				IRQ*			
Network Controller (TP)	IRQ*							
Basic & Modular Backplane 4 (TP)	USER1*	IEI*/USE R2* direct or cascade	IEO*/US ER3* direct or cascade	USER4	USER5	USER6	USER7	USER8
6809E/6309EP (TP)	FIRQ*	E*	RW*					
65C02 65C816 6803/6303 6808 6809/6309 68HC11 65C22 6840 65C21 (Alan Cox)	FIRQ* (some)	E*	RW*					
TMS9995 (Alan Cox)	MEMEN * ?	CRUIN * ?	CRUCLK * ?					

* = jumpered so the end user can select if the bus pin is connected or not

Product	Pin 37	Pin 38	Pin 39	Pin 40	Pin 77	Pin 78	Pin 79	Pin 80
PPI (Dino)	LED*	SLT_A*	CASS*	SLT_B*	OUT*			
Easy-Z80 (Sergey Kiselev)		IEI	IEO					
Z80Ctrl/CPU/IOX (JBLangston)	SCL	MISO	MOSI		IOXCS	AUXCS1	AUXCS2	

* = jumpered so the end user can select if the bus pin is connected or not

Appendix C - Bus Conventions For Mapping Motorola Busses

Introduction

The Motorola style 8-bit bus differs considerably from the bus expected by the conventional RCBus. It is possible to map from one to the other but it can be useful when integrating Motorola bus devices to make the Motorola bus signals available.

This appendix documents the existing conventions that are used for mapping a Motorola style bus to the RCBus. It is intended to be descriptive not prescriptive.

Mapping The Bus

The following signals are mapped directly onto the RCBus from the Motorola style bus

A15-A0: Address bus
D0-D7: Data bus
INT: IRQ (open collector)
RESET: RESET (see notes section)
CLOCK: (see notes section)

The Motorola bus has two different signals. E is a square wave clocking the bus. During one half of the E cycle the signals change, during the other half of the E cycle the signals are valid. There is no provision for an "idle" cycle, instead an additional read cycle is generated. This is usually targetted at a dummy location such as FFFF but some processors will generate dummy read cycles to other addresses and this can require care and is usually handled in software.

The second signal is RW. This indicates if the cycle is a read or a write using a single line unlike the 8080/Z80 bus where \RD and \RW may both be high to indicate no activity.

The Z80 style \RD and \WR signals are generated by combining the E clock with RW so that \RD or \WR goes low only when the bus state is valid.

As the Motorola bus has no notion of a separate I/O space an I/O window is normally used. By convention this is at 0xFEXX because this address window is suitable for existing operating systems for these platforms and mirrors many historic machines. There is no requirement to use 0xFEXX as the I/O card will generally only decode the low 8-bits of the address bus anyway.

The two signals for controlling the cycle type on the Z80 bus are \MREQ for a memory request and \IORQ for an I/O request. These can be generated by decoding the upper bits of the address generated by the processor when the bus is valid.

The final 40-pin RCBUS signal is /M1. This has no equivalent on Motorola bus processors as it is used as part of the Z80 interrupt decode not just as an indication of instruction start. The current cards pull this high so that the peripheral cards do not decode bus activity as a Z80 interrupt cycle.

Mapping The Extended Bus

The extended bus provides A23-A16, which are directly equivalent to A23-A16 on the 65C816 card.

The extended bus provides several signals that have no easy mapping. These are /BUSRQ /BUSACK and /HALT. None of these signals are used by most peripheral cards except specialist cards such as the Z80 DMA interface.

The other two signals mostly map. The \NMI signal is equivalent to the \NMI signal on Motorola bus systems (called XIRQ on some processors). The /WAIT signal is near enough the same semantics as the Z80 one that it can be provided except on the 6309E/6809E which do not support clock stretching this way as they are intended to run synchronously with a SAM or similar device on the other half of the E cycle.

Additional Signals

Some Motorola bus peripherals are complicated (or near impossible) to operate without the Motorola bus signals. At other times it is just useful to reduce chip count to have access.

Existing Motorola bus processor cards can provide the E RW and \FIRQ signals on bus pins. Jumpers should be used as the lines are intended to be available to the user for other purposes if desired.

37	\FIRQ	Open collector, pull up on CPU card
38	E	E clock
39	RW	RW signal from processor

Using these signals on a peripheral device makes the peripheral card incompatible with the basic RCbus. There is a trade-off between the convenience and simplicity of interfacing and the compatibility.

Notes

Reset

The reset signal on many classic RCbus boards is very poor. The original RC2014™ systems in particular lack a proper reset controller. The Motorola bus devices that need a clean reset (such as the 68HC11) should include their own reset controller to clean up the reset during power on.

Clock

The conventional RCbus clock was 7.372MHz. This is also conveniently a clock that generates good serial signals and a bit under 2MHz E clock for 63xx/68xx processors. There is no requirement to use this clock, but it does improve compatibility. For slower parts half this clock is similarly convenient.

The 6502 processor clock input and E clock are the same barring skew. This effectively means a 2MHz 6502 has the same timing requirements as the 7.37MHz Z80. Whilst the RCbus can be run with a high speed 65C02 or 65C816 part it will be necessary to use 74AHCT parts in general, and even then some of the standard boards such as the 512K/512K memory card will be too slow to go above about 8MHz.

Bus Hold

68xx and 63xx series devices have a bus hold time (the time that signals remain valid on the data and address bus after \WR rises) that is broadly compatible with the Z80 timings used on the bus. The "classic" 6502 and 65C02 parts likewise do. Modern 65C02 and 65C816 parts have almost no bus hold. On a backplane it becomes necessary for the processor card to cut the \WR signal early in order to produce a bus hold, otherwise many RCbus cards will not work.

Signal Bounce and Buffering

Some of the NMOS parts generate significant ground bounce when the address bus changes if they are driving a load with significant capacitance - such as an RCbus backplane. In these cases it may be wise to buffer the signals. Buffering signals from NMOS parts also improves compatibility with standard RCbus cards. This is not normally a problem when driving 74HCT series parts, but can be for driving other things directly (such as the CF adapter).

Memory Layout

The 63/68xx and 65xx parts require ROM is present at the top rather than bottom of memory at boot. The classic 512/512K card provides this, whilst the standard 'flat' 512/512K cards have a jumper to switch the RAM/ROM over.