CFT Minicomputer Bus Guide

Alexios Chouchoulas
alexios@bedroomlan.org

Abstract

This document discusses design and implementational details of the CFT backplane and bus. The design uses two separate buses, one for the processor and control logic, and one for memory and peripherals. The two buses are connected using a trivial bus bridge, as well as an optional interrupt controller and address space extension board to allow more than 65,536 words of memory.
A basic discussion of non-normative mechanical and electrical properties is included, as well as a description of the semantics, direction and behaviour of all signals on the buses and their relation to other signals.

History of Changes

2011-11-06 Initial revision.
2011-11-09 Clarified behaviour of reset mechanism. Froze definition of reset signals.
2011-11-10 Modified Control Bus pin-out. Added description for missing bus signals.
2011-11-15 Modified Control Bus pin-out (revision G). Removed OPIF3–OPIF0 signals. Added IR2–IR0. Corrected power pin-out. Corrected minor issue with pin-out descriptions.
2011-11-20 Modified Control Bus pin-out (revision H) to include FV signal.

1 Introduction

The CFT design incorporates two separate, bridged buses, both meant for 96-pin DIN 41612 'Bauform C' connectors and 100x160mm (3U) Eurocards (IEEE 1101).

The Control Bus provides an interconnect for signals local to the processor part of the computer. This includes the IBUS, signals from the Control Unit, and Microcode ROM, as well as signals to and from the ALU, register boards, interrupt controller boards, and all other processor components.

The Expansion Bus provides a computer-level expansion interface. It allows devices to connect to the memory space, I/O space, and expanded memory space by using the Address Bus, Data Bus, their associated control lines, as well as interrupt lines.

The two buses are connected using a two-slot bus bridge card which also implements the required expansion bus logic.

2 Backplane

The CFT backplane is based on a part available to the author at the time when the CFT was originally designed. Historically, it belonged to a computer used and possibly designed at the University of Edinburgh's Department of Computer Science. The machines were decommissioned circa 1993 and were based on a Motorola MC68020 microprocessor.

The backplane itself is a Eurocard (IEEE 1101) design using DIN 41612 connectors. The connectors are *Bauform* (style) 'C' 96-pin receptacles.

Five power rails are provided at the edges of the card, occupying a total of 6 rows, or 18 pins.

Of the remaining 78 pins, 2 pins per connector are

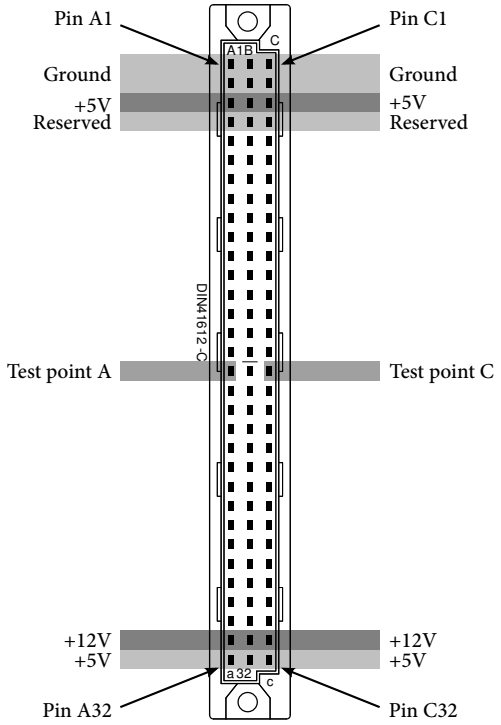

Figure 1: A card receptacle on the CFT backplane (as seen from the top) showing the power and ground rails including one reserved for future expansion, as well as the two local test point pins. All other pins are connected across the backplane and terminated at either end.

not bussed. Instead, they are broken out to test points on the backplane. This leaves 76 pins for bus signals.

All 76 signal pins are terminated with 470Ω pull-up resistors.

Both buses share the same physical backplane. To isolate them, the suggestion is to cut the traces between the last slot of the control bus and the first slot of the expansion bus, and to provide termination resistors on the bus bridge card.

If building the machine from scratch, two separate backplanes or DIN 41612 connectors on ribbon cables may be used, or the constructor may develop a custom *ad hoc* solution. In the interest of allowing implementations on other, possibly unterminated backplanes, optional bus termination will also be provided on two cards, one on the control side and one on the expansion side.

3 Control Bus

provides an interconnect for signals local to the processor part of the computer. This includes the IBUS, signals from the Control Unit, and Microcode ROM, as well as signals to and from the ALU, register boards, interrupt controller boards, and all other processor components.

Table 1: Control Bus Pin-out.

	A	B	C
1	GND	GND	GND
2	GND	GND	GND
3	+5V	+5V	+5V
4	Reserved	Reserved	Reserved
5	IR0	IR1	IBUS0
6	IR2	Reserved	IBUS1
7	$\overline{\text{CLI}}$	$\overline{\text{SKIP}}$	IBUS2
8	$\overline{\text{STI}}$	AINDEX	IBUS3
9	$\overline{\text{uINSTR18}}$	$\overline{\text{CLL}}$	IBUS4
10	$\overline{\text{HALT}}$	$\overline{\text{CPL}}$	IBUS5
11	$\overline{\text{END}}$	Reserved	IBUS6
12	$\overline{\text{IRQ}}$	PC10	IBUS7
13	$\overline{\text{IRQS}}$	PC11	$\overline{\text{MEM}}$
14	RUNIT0	PC12	$\overline{\text{IO}}$
15	RUNIT1	PC13	$\overline{\text{R}}$
16	RUNIT2	PC14	$\overline{\text{W}}$
17	TPA	PC15	TPC
18	RUNIT3	ISROLL	FL
19	$\overline{\text{WAC}}$	$\overline{\text{RAC}}$	FZERO
20	$\overline{\text{WALU}}$	$\overline{\text{RAGL}}$	FNEG
21	$\overline{\text{WDR}}$	$\overline{\text{RDR}}$	IBUS8
22	$\overline{\text{WIR}}$	$\overline{\text{RPC}}$	IBUS9
23	$\overline{\text{WMAR}}$	$\overline{\text{INCAC}}$	IBUS10
24	$\overline{\text{WPC}}$	$\overline{\text{INCDR}}$	IBUS11
25	$\overline{\text{SYSDEV}}$	$\overline{\text{INCPC}}$	IBUS12
26	FV	Reserved	IBUS13
27	Reserved	Reserved	IBUS14
28	$\overline{\text{GUARD}}$	CLK4	IBUS15
29	CLK5	CLK2	CLK1
30	$\overline{\text{RSTHOLD}}$	CLK3	$\overline{\text{RESET}}$
31	+12V	+12V	+12V
32	+5V	+5V	+5V

The following is a description of the signals of the control bus.

3.1 Power, Synchronisation and Initialisation

Power is supplied on four power rails, connected directly to the power supply.

GND (Pins A1–2, B1–2, C1–2) Connected to the power supply's ground rail.

+5V (Pins A3, B3, C3, A32, B32, C32) Connected to the power supply's +5V rail.

Reserved Power Rail (Pins A4, B4, C4) This is a power rail, although the exact voltage to be used has not

yet been decided. A likely candidate is +3.3V, which is a fairly common voltage for many devices.

+12V (Pins A31, B31, C31) Connected to the power supply's +12V rail.

Synchronisation and reset functionality is served by the following signals, all of which are outputs:

CLK1 (C29) is the main system clock phase, a 50% duty cycle square wave. The exact period of this clock is beyond the scope of this document and is purposefully left unspecified.

CLK2 (B29) is the second phase of the clock. It is a 50% duty cycle clock signal derived by shifting CLK1 by 90°.

CLK3 (B30) is the third phase of the clock. It is a 50% duty cycle clock signal derived by shifting CLK1 by 180°, which is tantamount to complementing CLK1.

CLK4 (B28) is the fourth phase of the clock. It is a 50% duty cycle clock signal derived by shifting CLK1 by 270°, or shifting CLK2 by 180°, or complementing CLK2.

CLK5 (A29) is a 75% duty cycle square wave. Its falling edge coincides with the falling edge of CLK1 and its rising edge coincides with the rising edge of CLK4, 90° and 25% of the period apart.

$\overline{\text{GUARD}}$ (A28) The guard pulse is a 25% (or less) duty cycle square wave, with its high level occurring just before the rising edge of CLK1. It helps resolve bus contentions between faster and slower units within the processor by forcing most units to disconnect from the IBUS when $\overline{\text{GUARD}}$ is high.

Please refer to figure 2 (p. 3) for a timing diagram of these clock phases.

Resetting the machine is handled by two signals, both of which are outputs. Devices may reset on the falling edge or low level of either or both of the two signals:

$\overline{\text{RESET}}$ (C30) When low, this short pulse indicates that a reset sequence has started. This signal may be driven low using an open collector to initiate a reset.

$\overline{\text{RSTHOLD}}$ (A30) When low, pulse indicates a reset is in progress. This pulse starts shortly after the rising edge of $\overline{\text{RESET}}$ and stays low for a relatively long time in terms of the machine's clock period.

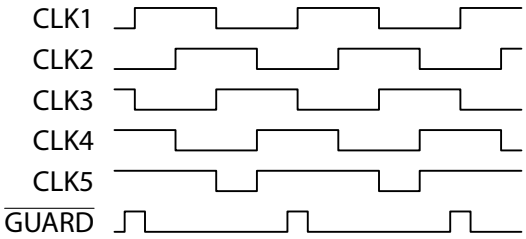

Figure 2: Clock timing diagram showing all the clock phases available to the Control Bus.

3.2 Unit Control Signals

These signals are all outputs from the microcode ROM. The microcode ROM outputs values whenever $\overline{\text{HALT}}$ is high. When $\overline{\text{HALT}}$ goes low, the ROM is tri-stated. Pull-up resistors pull these signals high, but other units (including the front panel) may drive the signals low to operate the CPU's units without the microcode ROM's intervention. This allows hardware expansions to the CFT processor to be implemented, if necessary.

$\overline{\text{CLI}}$ (A7) When low, this signal clears the Interrupt register, disabling subsequent interrupts.

$\overline{\text{STI}}$ (A8) When low, this signal sets the Interrupt register, enabling interrupts.

$\overline{\text{uINSTR18}}$ (A9) Bit 18 of the currently executing microinstruction. This is unused in the current versions of the microcode, but the signal is broken out onto the bus for future expansion. This signal is pulled up.

$\overline{\text{END}}$ (A11) Is low at the end of the current microprogram.

RUNIT3–RUNIT0 (A18, A16–A14) The corresponding unit is output to the IBUS. This group of signals is undecoded in the sequencer unit, since other units (including the ALU) need to perform their own decoding.

0000 Idle; no unit outputs to the IBUS.

0001 Reserved.

0010 AGL.

0011 PC.

0100 DR.

0101 AC.

1000 ALU: read from the adder unit.

1001 ALU: read from the bitwise AND unit.

1010 ALU: read from the bitwise OR unit.

1011 ALU: read from the bitwise XOR unit.

1100 ALU: read from the roll unit.

1101 ALU: read from the bitwise NOT unit.

1110 Constant Source, bank 1.

1111 Constant Source, bank 2.

$\overline{\text{WAC}}$ (A19) When low, this signal latches data from the IBUS into the Accumulator.

$\overline{\text{WALU}}$ (A20) On the *rising edge* of this signal, the ALU's Port B clocks data from the IBUS. (Port A is the Accumulator)

$\overline{\text{WDR}}$ (A21) When low, this signal latches data from the IBUS into the Data Register.

$\overline{\text{WIR}}$ (A22) The IR reads its value from the IBUS on the *rising edge* of this signal.

$\overline{\text{WMAR}}$ (A23) The MAR reads its value from the IBUS on the *rising edge* of this signal.

$\overline{\text{WPC}}$ (A24) When low, this signal latches data from the IBUS into the Program Counter.

IR2–IR0 (A6, B5, A5) Outputs the three least significant bits of the Instruction Register. This is used by the ALU.

$\overline{\text{CLL}}$ (B9) When low, this signal clears the Link register.

$\overline{\text{CPL}}$ (B10) Open Collector signal, input to the L register. When low, this signal complements (toggles) the Link register. Since both the ALU and control unit need to do this, an open collector circuit is mandated to avoid bus contention.

$\overline{\text{RAC}}$ (B19) This signal decodes RUNIT. It is low when RUNIT3–RUNIT0=0101, which outputs the Accumulator on the IBUS.

$\overline{\text{RAGL}}$ (B20) This signal decodes RUNIT. It is low when RUNIT3–RUNIT0=0010, which enables the value of the AGL on the IBUS.

$\overline{\text{RDR}}$ (B21) This signal decodes RUNIT. It is low when RUNIT3–RUNIT0=0100. This outputs the Data Register on the IBUS.

$\overline{\text{RPC}}$ (B22) This signal decodes RUNIT. It is low when RUNIT3–RUNIT0=0011. This outputs the Program Counter the IBUS.

$\overline{\text{INCAC}}$ (B23) The rising edge of this signal increments the Accumulator by one.

$\overline{\text{INCDR}}$ (B24) The rising edge of this signal increments the Data Register by one.

$\overline{\text{INCPC}}$ (B25) The rising edge of this signal increments the Program Counter by one.

$\overline{\text{MEM}}$ (C13) When low, this signal indicates that a memory read or write cycle is taking place. Memory and memory-mapped peripherals attached to the computer should respond to this signal by decoding the address bus and driving any chip select lines necessary to prepare the memory for I/O.

$\overline{\text{IO}}$ (C14) When low, this signal indicates that an I/O space read or write cycle is taking place. I/O peripherals should enable register sets and other mapped functionality in response to this signal.

$\overline{\text{R}}$ (C15) When low, indicates that a read is taking place. The exact timings of a memory or I/O space read cycle are purposefully left unspecified in this document.

$\overline{\text{W}}$ (C16) Like $\overline{\text{R}}$, when low, indicates that a write cycle is taking place.

3.3 Microcode Control Signals

These signals control the behaviour of the microcode ROM. They are inputs to the control unit and in almost all cases are driven full-time by another unit within the computer. Exceptions are noted.

$\overline{\text{HALT}}$ (A10) Input to the microcode unit, **open collector**. Driving this signal low stops the computer's processor. $\overline{\text{HALT}}$ inhibits the microprogram counter (μPC) and tri-states the microcode ROM. In this state, the computer stops executing microinstructions and hence instructions. Control signals will are held by a bus hold circuit, but may be overridden by other units as long as $\overline{\text{HALT}}$ is asserted. It may also be used by devices that do not intend to modify the processor's internal state as a means of getting simple, asynchronous wait states. The $\overline{\text{HALT}}$ signal *must* be driven by an open collector circuit to avoid contention with other devices asserting it.

$\overline{\text{IRQ}}$ (A12) Input to the processor, **open collector**. Asserting this signal notifies the control unit that of an interrupt request. The machine allows for only one, maskable interrupt. An interrupt controller may be used to provide multiple interrupt lines to the expansion bus (eight such lines, $\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ are

provided on the expansion bus). For more information, please consult section 4 (p. 5). Please note that this line must be driven using open collector circuitry to avoid bus contention.

$\overline{\text{IRQS}}$ (A13) Input to the microcode unit. The Interrupt Request Seen signal is driven at all times by the internal interrupt state machine and is fed directly to the microcode ROM. It is asserted at the end of an microprogram if $\overline{\text{IRQ}}$ was asserted during execution of the microprogram, and the I is set.

$\overline{\text{SKIP}}$ (B7) Input from the Skip and Branch Unit (SBU) to the microcode unit. The SBU asserts this signal when a skip or branch must be taken, which in turn branches the current microprogram accordingly. The behaviour of the SBU depends on the current value of the three flags (signals FL, FZERO, FNEG) and the value of the ten least significant bits of the IR. The resultant behaviour of the computer depends on the microcode being executed. The SBU asserts this signal at all times but most microprograms treat $\overline{\text{SKIP}}$ as a don't-care signal.

AINDEX (B8) Input from the autoindex decoder to the microcode unit. Based on the programming model, AINDEXis asserted when the MAR is loaded with a value in the range 0080–00FF. The signal remains asserted until the end of the currently executed microprogram, i.e. until $\overline{\text{END}}$ is asserted.

3.4 Other Signals

$\overline{\text{SYSDEV}}$ (A25) This signal simplifies adding I/O-mapped devices to the machine. It is low when MAR contains an address in the range 0000–00FF. The most significant 8 bits of MAR are fully decoded for this.

PC15–PC10 (B17–B12) the six most significant bits of the PC, output at all times. This is the current page of the processor, and is used by the AGL to generate addresses from instructions.

ISROLL (B18) Active high input from the SBU to the microcode unit. This signal is asserted when any of the least significant three bits of the IR are non-zero. If ISROLL is asserted and OPIF3–OPIF0 is 1011, the SBU notifies the microcode unit via the $\overline{\text{SKIP}}$ signal that a roll operation *may* be selected. Depending on the current microprogram, the microcode can then act on this information. **Nota Bene:** The semantics and direction of this signal are under review.

IBUS15–IBUS0 (Pins C28–C21 and C12–C5) Bi-directional, 16-bit Internal Bus. This is the primary means of communicating data within the CFT micro-architecture. Depending on the value of RUNIT3–RUNIT0 as well as $\overline{\text{MEM}}$, $\overline{\text{IO}}$ and $\overline{\text{R}}$, various units drive the IBUS. Most units should float (high impedance) their bus drivers whenever $\overline{\text{GUARD}}$ is asserted (low) to avoid bus contention with units that may drive the bus for longer than allowed (due to propagation delays, for instance).

FV (A26) Active-high output. The current value of the Overflow flag. This is an output from the ALU to the control unit.

FL (C18) Active-high output. The current value of the Link Register (L). This is an output from the ALU, currently used by the control unit.

FZERO (C19) Active-high output. FZERO is asserted whenever the value of the Accumulator (AC) is zero.

FNEG (C20) Active-high output. FNEG is asserted whenever the value of AC is negative. This is tantamount to bit 15 of the Accumulator being set.

3.5 Test Points

Every slot on the backplane has two local pins, designated TPA (pin A17) and TPC (pin C17). These are not connected to other slots (unbussed), but are broken out to test points and pins on the backplane. They may be used to provide two local signals for debugging or indicators, or simply for testing. The CFT design does not currently use them.

4 Expansion Bus

The Expansion Bus provides a computer-level expansion interface. It allows devices to connect to the memory space, I/O space, and expanded memory space by using the Address Bus, Data Bus, their associated control lines, as well as interrupt lines. This is meant for devices that do not require access to the inner workings of the processor and its micro-architecture such as memory, serial and parallel interfaces, disk adapters, and so on.

The expansion bus is designed to allow two different types of cards.

- Memory cards and exotic devices that might need access to the address bus expansion signals (such as framebuffers) need to use a 96-pin plug (rows

Table 2: Expansion Bus Pin-out. Row B is optional. Expansion cards that need no access to the banked memory scheme may use 64-pin (A+C) DIN 41612 connectors, which are more cost-effective.

	A	B	C
1	GND	GND	GND
2	GND	GND	GND
3	+5V	+5V	+5V
4	Reserved	Reserved	Reserved
5	AB0	Reserved	DB0
6	AB1	Reserved	DB1
7	AB2	Reserved	DB2
8	AB3	Reserved	DB3
9	AB4	Reserved	DB4
10	AB5	Reserved	DB5
11	AB6	Reserved	DB6
12	AB7	Reserved	DB7
13	$\overline{\text{IRQ3}}$	AEXT0	$\overline{\text{MEM}}$
14	$\overline{\text{IRQ4}}$	AEXT1	$\overline{\text{IO}}$
15	$\overline{\text{IRQ5}}$	AEXT2	$\overline{\text{R}}$
16	$\overline{\text{IRQ6}}$	AEXT3	$\overline{\text{W}}$
17	TPA	AEXT4	TPC
18	$\overline{\text{IRQ7}}$	AEXT5	$\overline{\text{IRQ0}}$
19	$\overline{\text{HALT}}$	AEXT6	$\overline{\text{IRQ1}}$
20	AB8	AEXT7	$\overline{\text{IRQ2}}$
21	AB9	Reserved	DB8
22	AB10	Reserved	DB9
23	AB11	Reserved	DB10
24	AB12	Reserved	DB11
25	AB13	Reserved	DB12
26	AB14	Reserved	DB13
27	AB15	Reserved	DB14
28	$\overline{\text{GUARD}}$	Reserved	DB15
29	CLKB	Reserved	CLKA
30	$\overline{\text{RSTHOLD}}$	Reserved	$\overline{\text{RESET}}$
31	+12V	+12V	+12V
32	+5V	+5V	+5V

A+B+C). Row B, the middle row, contains expanded signals and unassigned bus signals available for cards to use.

- Other peripheral cards can use 64-pin plugs (A+C). DIN-41612 allows 64-pin plugs to fit 96-pin sockets, and this reduces the cost of peripheral cards.

4.1 Power, Synchronisation and Initialisation

Most of the pins in this category are identical in position and function to those in the Control bus, dictated to a great extent by the physical layout of the backplane. Please refer to section 3.1 (p. 2) for more information.

Power is supplied on four power rails, connected directly to the power supply.

GND (Pins A1–2, B1–2, C1–2) Connected to the power supply's ground rail.

+5V (Pins A3, B3, C3, A32, B32, C32) Connected to the power supply's +5V rail.

Reserved Power Rail (Pins A4, B4, C4) This is a power rail, although the exact voltage to be used has not yet been decided. A likely candidate is +3.3V, which is a fairly common voltage for many devices.

+12V (Pins A31, B31, C31) Connected to the power supply's +12V rail.

Synchronisation and reset functionality is served by the following signals, all of which are outputs:

CLKA (C29) is identical to, directly connected to, and located on the same pin as the Control Bus signal CLK1. This is the main system clock phase, a 50% duty cycle square wave. The exact period of this clock is beyond the scope of this document and is purposefully left unspecified.

CLKB (A29) is identical to, directly connected to, and located on the same pin as the Control Bus signal CLK5. This is a 75% duty cycle square wave. Its falling edge coincides with the falling edge of CLK1 and its rising edge coincides with the rising edge of CLK4, 90° and 25% of the period apart.

$\overline{\text{GUARD}}$ (A28) The guard pulse is a 25% (or less) duty cycle square wave, with its high level occurring just before the rising edge of CLK1. It helps resolve bus contentions between faster and slower units within the processor by forcing most units to disconnect from the IBUS when $\overline{\text{GUARD}}$ is high.

Please refer to figure 2 (p. 3) for a timing diagram of these clock phases, and the additional clock phases used only internally by the control logic.

Resetting the machine is handled by two signals, both of which are outputs. Devices may reset on the falling edge or low level of either or both of the two signals:

$\overline{\text{RESET}}$ (C30) When low, this short pulse indicates that a reset sequence has started. This signal may be driven low using an open collector to initiate a reset.

$\overline{\text{RSTHOLD}}$ (A30) When low, pulse indicates a reset is in progress. This pulse starts shortly after the rising edge of $\overline{\text{RESET}}$ and stays low for a relatively long time in terms of the machine's clock period.

4.2 Address, Data and Control Signals

Devices can use these signals to attach to the computer using a very simple bus protocol with timings identical to those used for static RAMs. The signals, which, to allow for more complex decoding tasks, are somewhat different than RAM signals, are:

AB15–AB0 (Pins A27–A20 and A12–A5) Output only, 16-bit Address Bus. This is used by the processor to specify a location in memory or I/O space to be written to or read from. The Address Bus is always driven by the processor's MAR register[1], so devices may never drive it.

AEXT7–AEXT0 (Pins B20–B13) Output only. These 8 lines extend the address space of the CFT architecture by 5 bits, to allow for up to 2,097,152 words to be accessed. The technique used is banking. Memory cards and memory-mapped devices exposing large blocks of memory (such as framebuffers) must use AEXT7-AEXT0 and AB12–AB0. If the device only uses 16 bits of memory, it must use AEXT2–AEXT0 and AB12–AB0. To avoid potential clashes between banked and unbanked memory, memory cards and memory-mapped devices should not use AB15–AB13. Such devices must, by necessity, use 96-pin (row A+B+C) DIN 41612 connectors.

DB15–DB0 (Pins C28–C21 and C12–C5) Bidirectional, 16-bit Data Bus. This is used to transfer data between the processor and devices. The Data Bus is connected to the processor's Internal Bus by a bus transceiver circuit when any of $\overline{\text{MEM}}$, $\overline{\text{IO}}$, $\overline{\text{R}}$ or $\overline{\text{W}}$ are asserted (they are asserted in pairs of $\overline{\text{MEM}}$ or $\overline{\text{IO}}$ and $\overline{\text{R}}$ or $\overline{\text{W}}$). Devices **must not** drive the data bus unless $\overline{\text{R}}$ and either $\overline{\text{MEM}}$ or $\overline{\text{IO}}$ (depending on device application) are asserted, and they have decoded the value of the Address Bus. Devices should not read from the data bus unless $\overline{\text{W}}$ and either $\overline{\text{MEM}}$ or $\overline{\text{IO}}$ are asserted and they have decoded the value of the Address Bus (some hardware, such as the front panel interface, may ignore this rule).

$\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ (Pins A18, A16–A13, C20–C18) Open collector inputs. Actual behaviour depends on the interrupt controller installed. At its simplest, all of these signals are electrically connected to the $\overline{\text{IRQ}}$ signal in the control logic, which is an open collector input, hence the open collector designation here. Even with a full-blown interrupt controller, however, having open collector outputs facilitates IRQ line sharing without glitches or bus contention[2].

$\overline{\text{MEM}}$ (C13) When low, this signal indicates that a memory read or write cycle is taking place. Memory and memory-mapped peripherals attached to the computer should respond to this signal by decoding the address bus and driving any chip select lines necessary to prepare the memory for I/O.

$\overline{\text{IO}}$ (C14) When low, this signal indicates that an I/O space read or write cycle is taking place. I/O peripherals should enable register sets and other mapped functionality in response to this signal.

$\overline{\text{R}}$ (C15) When low, indicates that a read is taking place. The exact timings of a memory or I/O space read cycle are purposefully left unspecified in this document.

$\overline{\text{W}}$ (C16) Like $\overline{\text{R}}$, when low, indicates that a write cycle is taking place.

4.3 Test Points

Every slot on the backplane has two local pins, designated TPA (pin A17) and TPC (pin C17). These are not connected to other slots (unbussed), but are broken out to test points and pins on the backplane. They may be used to provide two local signals for debugging or indicators, or simply for testing. The CFT design does not currently use them.

5 Bus Bridge

The purpose of the bus bridge is to connect (electrically or logically) signals on the Control Bus to signals on the Expansion Bus. Its functions are:

[1] This makes Bus Mastering and Direct Memory Access (DMA) impossible, but it is also unnecessary for the CFT.

[2] This feature alone makes the CFT computer better than the original IBM PC, which was plagued for years by improperly designed interrupt mechanisms.

- Connecting the outputs of the Memory Address Register (MAR) to the Address Bus.
- Connecting, when required, the IBUS to the Data Bus via a bus transceiver.
- Connecting directly the various control signals that are shared between the Control and Expansion buses.
- Providing Address Space Extension signals to the Expansion Bus.
- Mapping the $\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$ interrupt request lines to the control unit's $\overline{\text{IRQ}}$ signal.

5.1 Address Space Extension

The CFT architecture has a 16-bit address space, which allows for 65,536 words of memory or I/O to be addressed. To facilitate accessing more memory, an Address Space Extension scheme is used. This provides for an additional 8 signals in row B of the Expansion Bus, namely AEXT7–AEXT0.

An address extension card may drive these signals in one of many ways for the benefit of memory cards.

Memory cards make up a 21-bit physical address using AEXT7–AEXT0 for the most significant 8 bits, and AB12–AB0 for the least significant 13 bits. This allows for a maximum of 2,097,152 words to be accessed, although this depends on the address extension scheme implemented.

There are two options at this point:

- A trivial, low-cost mapping, by grounding AEXT7–AEXT3 and connecting AEXT2–AEXT0 to AB15–AB13 respectively. This provides a one-to-one mapping between the control unit's address space and the physical address space. Up to 65,536 words of memory or memory-mapped devices are accessible to the processor.
- A banked memory controller, the specifics of which are beyond the scope of this document. As AEXT extends the address space by 5 bits, up to 21 bits of memory (or 2,097,152 words) may be accessed. Each bank contains 8,192 words. The processor's address space is broken into 8 banks. How these 8 banks map to physical memory depends on the controller used.

5.2 Interrupt Request Line Mapping

The Expansion Bus specifies 8 Interrupt Request Lines, $\overline{\text{IRQ7}}$–$\overline{\text{IRQ0}}$. The control unit accommodates a single interrupt signal, $\overline{\text{IRQ}}$.

One of the duties of the bus bridge is to provide a connection or mapping between these eight lines and the control unit's one. There are many ways this can be done. The two most obvious ones are as follows:

- The trivial solution is to directly, electrically connect all $\overline{\text{IRQ}}$n lines to $\overline{\text{IRQ}}$. This is possible because these are all open collector signals and bus contention is not an issue. Detecting which device caused an interrupt is left to the computer's control program.
- An interrupt controller which collects the 8 interrupt request lines, possibly masking some of them, and sends an interrupt request signal to the processor. The computer's control program can query the interrupt controller card to find out which interrupt lines have been triggered, and act accordingly. This delegates some of the interrupt processing to hardware, and is almost certainly faster than the trivial interrupt logic.

Either solution may be implemented. Additional, hybrid solutions may also need to be implemented by hardware that needs additional interrupt lines or functionality.

6 Further reading

For further information on the CFT computer, please consult the following URL:

www.bedroomlan.org/hardware/cft

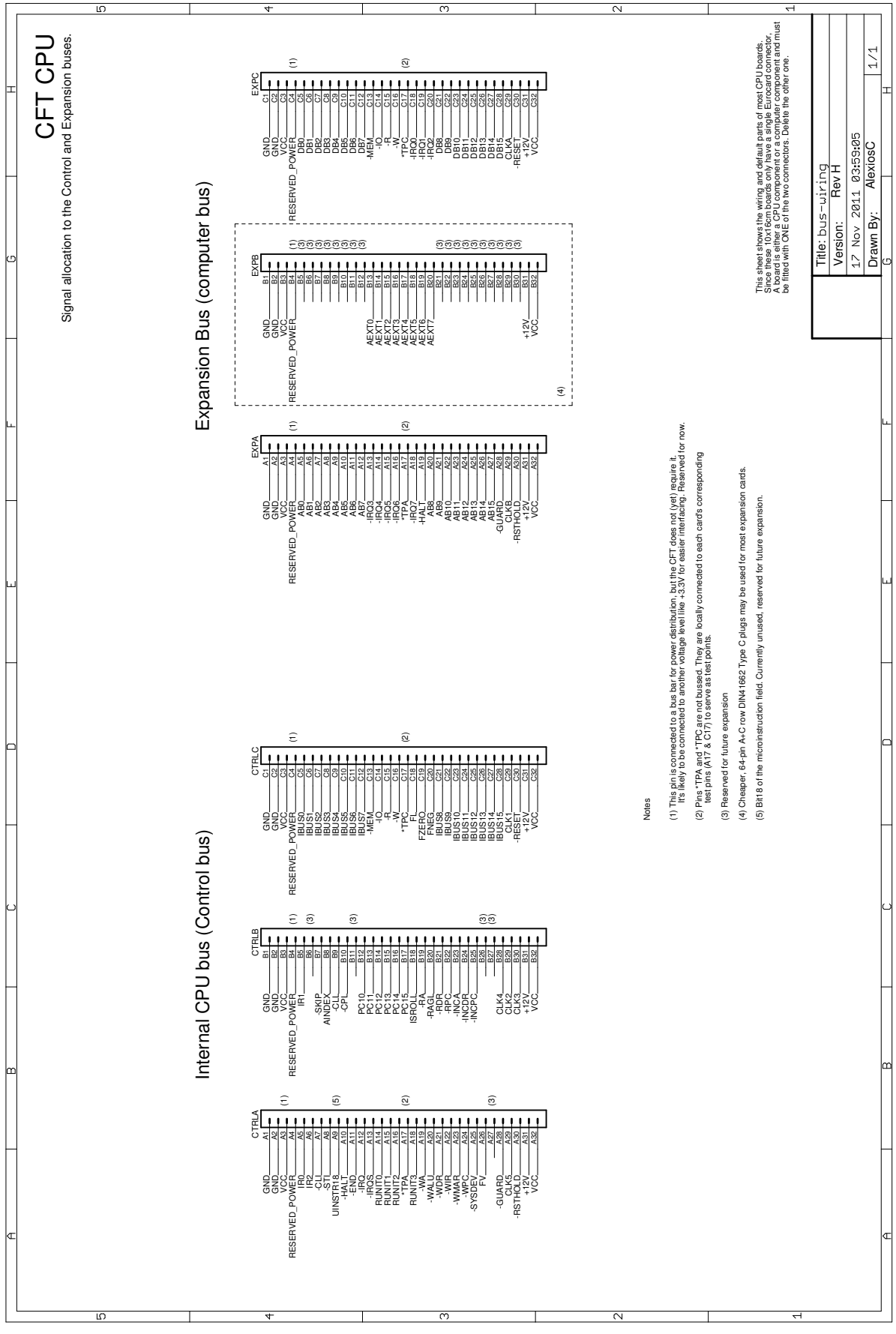

Figure 3: Schematic drawing of the bus wiring.