The P32W Bus

The backplane: the bus cycle, mastership, interrupts, every pin and signal

P32W — Ken Rother's 32-bit TTL computer

Generated 2026-11-13 from the repository at 3cacc22 2026-11-13

Contents

The P32W Bus `docs/bus.md`

- The bus cycle
- Bus mastership (DMA)
- Interrupts
- Slots
- J1
- J2
- J3
- Signals

The backplane `docs/backplane.md`

1. What it holds
2. The parts no card can hold
3. Verification
4. The board (D79)

The backplane: theory of operation `docs/backplane-theory.md`

1. What the backplane does
2. The slots
3. Bused lines and per-slot lines
4. Requests, grants and the lock
5. The pull-ups
6. The clock
7. Interrupts and the acknowledge cycle
8. Power
9. An empty slot
10. Verified
11. Open questions

The P32W Bus

docs/bus.md

Status: IN USE: every card's netlist and the backplane are generated from this definition and checked against it (D50, D52, D57, D59, D64, D68, D74, D76; last change 2026-11-11, D102: -CPUIN and -MMUIN on two of J3's spare pins).

Generated by `generators/p32bus.py` *— edit the generator, not this file.*

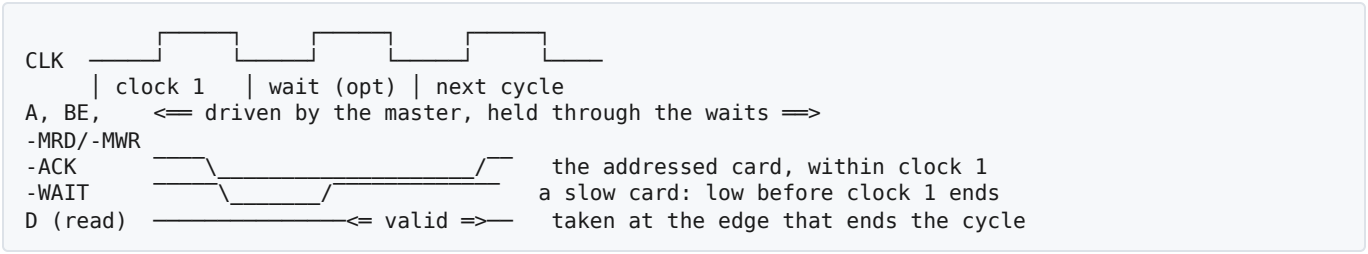
Three 96-pin DIN 41612 connectors per card (D49), down the 366.7 mm edge of a 9U × 340 mm card: **J1** the bus cycle (data, physical address, byte enables, strobes, `-ACK`, `-WAIT`), **J2** system control (clock, reset, interrupts, arbitration), **J3** the CPU ↔ MMU/data local link in slots 1 and 2 and the debug slot 0 (power only in the others). Pins 1–2 of every row are VCC, 31–32 GND; row B odd pins 3–29 are GND (a guard between rows A and C); J2's row B even pins 4–30 are VCC as well (D57), so J2 carries alternate supply pins down its middle. Active-low signals have a leading dash; `-ACK`, `-WAIT`, `-LOCK` and `-IRQn` are open-drain with backplane pull-ups. Spare pins on J1 and J2 are bused to every slot (D68); J3's eight (B28, B30, C25–C30) join slots 0, 1 and 2, as the local link does (D99): spare private lines. The backplane itself: backplane.md.

The backplane does **not** carry the CPU's microcode fields: only physical memory cycles, so any card can be a bus master (DMA), any number of memory cards can answer, and a slow card can stretch a cycle.

Pins (per card): 171 signal, 32 VCC, 60 GND, 25 spare (of 288).

The bus cycle

The bus is **synchronous**: one clock (`CLK`, the CPU's microcycle: 3 MHz, or 4 MHz with the CPU card's clock jumper, D72, D74) for every card. A cycle is one clock, stretched by `-WAIT`:



1. **The master** drives `A2–A31`, `BE0–BE3` and `-MRD` or `-MWR` from the start of the clock (on a write, `D` too).
2. **The addressed card** pulls `-ACK` low within the first clock: address decoding is fast and always fits. No `-ACK` by the end of the first clock is a **BUSERROR**, so a missing card can never hang the bus.
3. **A slow card** also pulls `-WAIT` low before the clock ends. The master holds everything for another clock, and checks `-WAIT` again; when it is high at the end of a clock, that edge ends the cycle (a read's data is taken then; a write is complete).
4. **The CPU** sees a waiting cycle as `STALL` on the local link: its clock `CCLK` has no edge until the cycle ends, so nothing on the CPU side changes. The microcode does not change for slower cards.

Bus mastership (DMA)

The **arbiter is on the MMU / data card** (slot 2), which is the CPU's own master interface. Any card in slots 3–10 may become master:

- It pulls its radial `-BREQ` low. The arbiter grants at a cycle boundary, when the CPU's next microcycle does not use the bus (most do not: about two in six are memory cycles), and holds the CPU (`STALL`) only when the CPU wants the bus while a device has it.
- `-BGNT` low means the card is master **from the next clock**. It runs as many cycles as it likes and gives the bus back by releasing `-BREQ`; the arbiter then drops `-BGNT`.
- **Priority:** fixed, among the requests present at the boundary: slot 0 (the debug card, D64) first, then the higher slot first.
- `-LOCK`: a master that holds `-LOCK` low keeps the bus for the next cycle too; the CPU uses it for an atomic read-modify-write (`LOCKREQ` on the local link), and the arbiter grants nothing while it is low.
- A DMA master addresses **physical** memory; it has no MMU. The kernel gives it physical buffers (and pins their pages while the transfer runs).

Interrupts

- `-IRQ0`–`-IRQ7`, open-drain levels into the CPU's `IMASK` / `IPEND` logic; the lowest-numbered pending, enabled level is taken first (`IRQ0`, the timer, has the highest priority).
- Each card's **jumpers** choose the level each of its devices drives (D52). Nothing about an interrupt depends on the slot a card is in.
- **Vectored (D57).** When the CPU takes level *n*, the MMU / data card runs an **interrupt acknowledge cycle**: a bus cycle like a read, with `-IACK` low instead of `-MRD` and `A4:A2` = *n* (the other address bits and `BE0`–`BE3` don't matter). The card driving level *n* pulls `-ACK` and puts its **vector** on `D7:D0`: its window number (D52) × 16 + the number of its device that is asking (its interrupt status register's bit; the lowest pending one on that level). A level is driven by **one card** — the jumpers of two cards must not share one — though one card may drive several. No `-ACK` (no card, or one without vectors) is not a bus error: the vector is `0xFF`, and the kernel reads the cards' interrupt status registers instead. The CPU puts the vector into `CAUSE` bits 15:8.

Slots

Slot	Card	J3
0	the debug card (D64): every signal, to listen or drive	the local link, a third drop
1	CPU	the local link
2	MMU / data (and the arbiter)	the local link
3–10	any: memory, disk, I/O, video	power only

The backplane has ten positions, slots 0–9, 1.5 inches apart (D99, docs/backplane.md): the bus's slot 10 has no socket there, and its request pair is unused. Cards go in with their components facing slot 0's end, the front.

Slot 0 carries everything: J1 and J2 as every slot, the radial `-BREQ3–10` / `-BGNT3–10` as slot 2 sees them (to watch or play every requester), its own pair `-BREQ0` / `-BGNT0` (the arbiter's highest priority), and J3. The debug card there can watch a running CPU's local link, or, with the CPU card pulled, play the CPU's side of it.

Every card is placed by its **jumpers**, not its slot (D50, D52): memory cards choose a 16 MiB window, device cards a 4 KiB window among the device registers, and their interrupt levels. Apart from slots 1 and 2 (the local link), any card works in any slot; the slot only sets its bus-request priority.

J1

Pin	Row A	Row B	Row C
1	VCC	VCC	VCC
2	VCC	VCC	VCC
3	D0	GND	D28
4	D1	A26	D29
5	D2	GND	D30
6	D3	A27	D31
7	D4	GND	A2
8	D5	A28	A3
9	D6	GND	A4
10	D7	A29	A5
11	D8	GND	A6
12	D9	A30	A7
13	D10	GND	A8
14	D11	A31	A9
15	D12	GND	A10
16	D13	BE0	A11
17	D14	GND	A12
18	D15	BE1	A13
19	D16	GND	A14
20	D17	BE2	A15
21	D18	GND	A16
22	D19	BE3	A17
23	D20	GND	A18
24	D21	-MRD	A19
25	D22	GND	A20
26	D23	-MWR	A21
27	D24	GND	A22
28	D25	-ACK	A23
29	D26	GND	A24
30	D27	-WAIT	A25
31	GND	GND	GND
32	GND	GND	GND

J2

Pin	Row A	Row B	Row C
1	VCC	VCC	VCC

Pin	Row A	Row B	Row C
2	VCC	VCC	VCC
3	CLK	GND	-BREQ3
4	-CLK	VCC	-BREQ4
5	-RES	GND	-BREQ5
6	PWROK	VCC	-BREQ6
7	-LOCK	GND	-BREQ7
8	-IRQ0	VCC	-BREQ8
9	-IRQ1	GND	-BREQ9
10	-IRQ2	VCC	-BREQ10
11	-IRQ3	GND	-BGNT3
12	-IRQ4	VCC	-BGNT4
13	-IRQ5	GND	-BGNT5
14	-IRQ6	VCC	-BGNT6
15	-IRQ7	GND	-BGNT7
16	-BREQ	VCC	-BGNT8
17	-BGNT	GND	-BGNT9
18	-IACK	VCC	-BGNT10
19	-FAST	GND	-BREQ0
20	XCLK	VCC	-BGNT0
21	-XCLKEN	GND	SPARE_J2C21
22	SPARE_J2A22	VCC	SPARE_J2C22
23	SPARE_J2A23	GND	SPARE_J2C23
24	SPARE_J2A24	VCC	SPARE_J2C24
25	SPARE_J2A25	GND	SPARE_J2C25
26	SPARE_J2A26	VCC	SPARE_J2C26
27	SPARE_J2A27	GND	SPARE_J2C27
28	SPARE_J2A28	VCC	SPARE_J2C28
29	SPARE_J2A29	GND	SPARE_J2C29
30	SPARE_J2A30	VCC	SPARE_J2C30
31	GND	GND	GND
32	GND	GND	GND

J3

Pin	Row A	Row B	Row C
1	VCC	VCC	VCC
2	VCC	VCC	VCC
3	LR0	GND	LR28

Pin	Row A	Row B	Row C
4	LR1	FAULT	LR29
5	LR2	GND	LR30
6	LR3	FC0	LR31
7	LR4	GND	RSRC0
8	LR5	FC1	RSRC1
9	LR6	GND	RSRC2
10	LR7	FC2	RSRC3
11	LR8	GND	RDST0
12	LR9	FC3	RDST1
13	LR10	GND	RDST2
14	LR11	-WSTB	MEM0
15	LR12	GND	MEM1
16	LR13	STALL	MEM2
17	LR14	GND	SIZE0
18	LR15	LOCKREQ	SIZE1
19	LR16	GND	SIGNED
20	LR17	CCLK	CRSEL0
21	LR18	GND	CRSEL1
22	LR19	PCINC	CRSEL2
23	LR20	GND	SUPER
24	LR21	MARPC	TLBW
25	LR22	GND	SPARE_J3C25
26	LR23	MARINC	SPARE_J3C26
27	LR24	GND	SPARE_J3C27
28	LR25	-CPUIN	SPARE_J3C28
29	LR26	GND	SPARE_J3C29
30	LR27	-MMUIN	SPARE_J3C30
31	GND	GND	GND
32	GND	GND	GND

Signals

Signal	Pin	Wired to	Driven by	Read by	Meaning
`D0`	J1 A3	every slot	the master (write) or the addressed card (read)	all	data bus
`D1`	J1 A4	every slot	the master (write) or the addressed card (read)	all	data bus
`D2`	J1 A5	every slot	the master (write) or the addressed card (read)	all	data bus

Signal	Pin	Wired to	Driven by	Read by	Meaning
`D3`	J1 A6	every slot	the master (write) or the addressed card (read)	all	data bus
`D4`	J1 A7	every slot	the master (write) or the addressed card (read)	all	data bus
`D5`	J1 A8	every slot	the master (write) or the addressed card (read)	all	data bus
`D6`	J1 A9	every slot	the master (write) or the addressed card (read)	all	data bus
`D7`	J1 A10	every slot	the master (write) or the addressed card (read)	all	data bus
`D8`	J1 A11	every slot	the master (write) or the addressed card (read)	all	data bus
`D9`	J1 A12	every slot	the master (write) or the addressed card (read)	all	data bus
`D10`	J1 A13	every slot	the master (write) or the addressed card (read)	all	data bus
`D11`	J1 A14	every slot	the master (write) or the addressed card (read)	all	data bus
`D12`	J1 A15	every slot	the master (write) or the addressed card (read)	all	data bus
`D13`	J1 A16	every slot	the master (write) or the addressed card (read)	all	data bus
`D14`	J1 A17	every slot	the master (write) or the addressed card (read)	all	data bus
`D15`	J1 A18	every slot	the master (write) or the addressed card (read)	all	data bus
`D16`	J1 A19	every slot	the master (write) or the addressed card (read)	all	data bus
`D17`	J1 A20	every slot	the master (write) or the addressed card (read)	all	data bus
`D18`	J1 A21	every slot	the master (write) or the addressed card (read)	all	data bus
`D19`	J1 A22	every slot	the master (write) or the addressed card (read)	all	data bus
`D20`	J1 A23	every slot	the master (write) or the addressed card (read)	all	data bus
`D21`	J1 A24	every slot	the master (write) or the addressed card (read)	all	data bus
`D22`	J1 A25	every slot	the master (write) or the addressed card (read)	all	data bus
`D23`	J1 A26	every slot	the master (write) or the addressed card (read)	all	data bus
`D24`	J1 A27	every slot	the master (write) or the addressed card (read)	all	data bus
`D25`	J1 A28	every slot	the master (write) or the addressed card (read)	all	data bus
`D26`	J1 A29	every slot	the master (write) or the addressed card (read)	all	data bus
`D27`	J1 A30	every slot	the master (write) or the addressed card (read)	all	data bus

Signal	Pin	Wired to	Driven by	Read by	Meaning
`D28`	J1 C3	every slot	the master (write) or the addressed card (read)	all	data bus
`D29`	J1 C4	every slot	the master (write) or the addressed card (read)	all	data bus
`D30`	J1 C5	every slot	the master (write) or the addressed card (read)	all	data bus
`D31`	J1 C6	every slot	the master (write) or the addressed card (read)	all	data bus
`A2`	J1 C7	every slot	the master	all	physical address bits 31:2
`A3`	J1 C8	every slot	the master	all	physical address bits 31:2
`A4`	J1 C9	every slot	the master	all	physical address bits 31:2
`A5`	J1 C10	every slot	the master	all	physical address bits 31:2
`A6`	J1 C11	every slot	the master	all	physical address bits 31:2
`A7`	J1 C12	every slot	the master	all	physical address bits 31:2
`A8`	J1 C13	every slot	the master	all	physical address bits 31:2
`A9`	J1 C14	every slot	the master	all	physical address bits 31:2
`A10`	J1 C15	every slot	the master	all	physical address bits 31:2
`A11`	J1 C16	every slot	the master	all	physical address bits 31:2
`A12`	J1 C17	every slot	the master	all	physical address bits 31:2
`A13`	J1 C18	every slot	the master	all	physical address bits 31:2
`A14`	J1 C19	every slot	the master	all	physical address bits 31:2
`A15`	J1 C20	every slot	the master	all	physical address bits 31:2
`A16`	J1 C21	every slot	the master	all	physical address bits 31:2
`A17`	J1 C22	every slot	the master	all	physical address bits 31:2
`A18`	J1 C23	every slot	the master	all	physical address bits 31:2
`A19`	J1 C24	every slot	the master	all	physical address bits 31:2
`A20`	J1 C25	every slot	the master	all	physical address bits 31:2
`A21`	J1 C26	every slot	the master	all	physical address bits 31:2
`A22`	J1 C27	every slot	the master	all	physical address bits 31:2
`A23`	J1 C28	every slot	the master	all	physical address bits 31:2
`A24`	J1 C29	every slot	the master	all	physical address bits 31:2
`A25`	J1 C30	every slot	the master	all	physical address bits 31:2
`A26`	J1 B4	every slot	the master	all	physical address bits 31:2
`A27`	J1 B6	every slot	the master	all	physical address bits 31:2
`A28`	J1 B8	every slot	the master	all	physical address bits 31:2
`A29`	J1 B10	every slot	the master	all	physical address bits 31:2
`A30`	J1 B12	every slot	the master	all	physical address bits 31:2
`A31`	J1 B14	every slot	the master	all	physical address bits 31:2
`BE0`	J1 B16	every slot	the master	all	byte enables (lanes 0-3; lane n = D[8n+7:8n])
`BE1`	J1 B18	every slot	the master	all	byte enables (lanes 0-3; lane n = D[8n+7:8n])
`BE2`	J1 B20	every slot	the master	all	byte enables (lanes 0-3; lane n = D[8n+7:8n])
`BE3`	J1 B22	every slot	the master	all	byte enables (lanes 0-3; lane n = D[8n+7:8n])

Signal	Pin	Wired to	Driven by	Read by	Meaning
`-MRD`	J1 B24	every slot	the master	all	read cycle: the addressed card drives D
`-MWR`	J1 B26	every slot	the master	all	write cycle: the addressed card takes D (enabled lanes)
`-ACK`	J1 B28	every slot	the addressed card	the master	open-drain: a card decodes A, in the cycle's first clock; none -> BUSERROR
`-WAIT`	J1 B30	every slot	the addressed card	the master	open-drain: stretch this cycle by one clock while low (a wait state)
`CLK`	J2 A3	every slot	CPU	all	the system clock (the microcycle) and its inverse
`-CLK`	J2 A4	every slot	CPU	all	the system clock (the microcycle) and its inverse
`-RES`	J2 A5	every slot	CPU	all	reset
`PWROK`	J2 A6	every slot	backplane	CPU	power good (the backplane's +5 V supervisor, D78); low = reset, falling = power failing
`-LOCK`	J2 A7	every slot	the master	arbiter	open-drain: keep the bus for the next cycle too (an atomic read-modify-write)
`-IRQ0`	J2 A8	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ1`	J2 A9	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ2`	J2 A10	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ3`	J2 A11	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ4`	J2 A12	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ5`	J2 A13	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ6`	J2 A14	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IRQ7`	J2 A15	every slot	devices (and the debug card in DRIVE, D101)	CPU	open-drain interrupt request levels (IRQ0 = timer, the highest priority: the lowest-numbered is taken first)
`-IACK`	J2 A18	every slot	the master (the MMU / data card for the CPU)	devices	interrupt acknowledge cycle (D57): instead of -MRD, with A4:A2 = the level; the card driving that level answers -ACK and puts its vector on D7:0
`XCLK`	J2 A20	every slot	the debug card (slot 0)	CPU	an external oscillator for the CPU card (D76): the debug card's 12 MHz, or one edge per host STEP; the CPU card takes it while -XCLKEN is low
`-XCLKEN`	J2 A21	every slot	the debug card (slot 0)	CPU	low: the CPU card's oscillator is XCLK (D76); with no debug card the CPU card's pull-up holds it high
`-FAST`	J2 A19	every slot	the CPU card's clock jumper (fitted = low); the debug card alone (D101)	CPU, I/O	low: the 4 MHz clock (D74), the I/O card's timer dividing by 4; open (the backplane's pull-up): 3 MHz. Set with the power off
`-BREQ`	J2 A16	slots 0, 3–10 (radial)	the card	arbiter	this slot asks for the bus (radial)
`-BGNT`	J2 A17	slots 0, 3–10 (radial)	arbiter	the card	this slot is master from the next clock (radial)
`-BREQ3`	J2 C3	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ4`	J2 C4	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ

Signal	Pin	Wired to	Driven by	Read by	Meaning
`-BREQ5`	J2 C5	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ6`	J2 C6	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ7`	J2 C7	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ8`	J2 C8	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ9`	J2 C9	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BREQ10`	J2 C10	slots 2 and 0 (radial)	slot n's card	arbiter	slot n's -BREQ
`-BGNT3`	J2 C11	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT4`	J2 C12	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT5`	J2 C13	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT6`	J2 C14	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT7`	J2 C15	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT8`	J2 C16	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT9`	J2 C17	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BGNT10`	J2 C18	slots 2 and 0 (radial)	arbiter (the debug card alone, D101)	slot n's card	slot n's -BGNT
`-BREQ0`	J2 C19	slots 2 and 0 (radial)	the debug card (slot 0)	arbiter	slot 0's -BREQ: the highest priority (D64)
`-BGNT0`	J2 C20	slots 2 and 0 (radial)	arbiter	the debug card (slot 0)	slot 0's -BGNT
`LR0`	J3 A3	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR1`	J3 A4	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR2`	J3 A5	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR3`	J3 A6	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR4`	J3 A7	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR5`	J3 A8	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in

Signal	Pin	Wired to	Driven by	Read by	Meaning
`LR6`	J3 A9	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR7`	J3 A10	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR8`	J3 A11	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR9`	J3 A12	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR10`	J3 A13	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR11`	J3 A14	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR12`	J3 A15	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR13`	J3 A16	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR14`	J3 A17	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR15`	J3 A18	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR16`	J3 A19	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR17`	J3 A20	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR18`	J3 A21	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR19`	J3 A22	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR20`	J3 A23	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR21`	J3 A24	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR22`	J3 A25	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in

Signal	Pin	Wired to	Driven by	Read by	Meaning
`LR23`	J3 A26	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR24`	J3 A27	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR25`	J3 A28	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR26`	J3 A29	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR27`	J3 A30	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR28`	J3 C3	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR29`	J3 C4	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR30`	J3 C5	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`LR31`	J3 C6	slots 0–2 only	CPU or MMU/data (RSRC)	CPU, MMU/data	the CPU's result bus R, extended to the MMU / data card: addresses to MAR, store data out, load data (steered), BADADDR and the MMU's control registers in
`RSRC0`	J3 C7	slots 0–2 only	CPU	MMU/data	R's source, encoded (microword RSRC)
`RSRC1`	J3 C8	slots 0–2 only	CPU	MMU/data	R's source, encoded (microword RSRC)
`RSRC2`	J3 C9	slots 0–2 only	CPU	MMU/data	R's source, encoded (microword RSRC)
`RSRC3`	J3 C10	slots 0–2 only	CPU	MMU/data	R's source, encoded (microword RSRC)
`RDST0`	J3 C11	slots 0–2 only	CPU	MMU/data	R's destination, encoded (microword RDST)
`RDST1`	J3 C12	slots 0–2 only	CPU	MMU/data	R's destination, encoded (microword RDST)
`RDST2`	J3 C13	slots 0–2 only	CPU	MMU/data	R's destination, encoded (microword RDST)
`MEM0`	J3 C14	slots 0–2 only	CPU	MMU/data	memory cycle type (microword MEM)
`MEM1`	J3 C15	slots 0–2 only	CPU	MMU/data	memory cycle type (microword MEM)
`MEM2`	J3 C16	slots 0–2 only	CPU	MMU/data	memory cycle type (microword MEM)
`SIZE0`	J3 C17	slots 0–2 only	CPU	MMU/data	access size and sign-extension (from IR)
`SIZE1`	J3 C18	slots 0–2 only	CPU	MMU/data	access size and sign-extension (from IR)

Signal	Pin	Wired to	Driven by	Read by	Meaning
SIGNED	J3 C19	slots 0–2 only	CPU	MMU/data	access size and sign-extension (from IR)
SUPER	J3 C23	slots 0–2 only	CPU	MMU/data	STATUS.S: 1 = supervisor (untranslated)
TLBW	J3 C24	slots 0–2 only	CPU	MMU/data	write TLBHI/TLBLO into the TLB
CRSEL0	J3 C20	slots 0–2 only	CPU	MMU/data	hardware control register for RSRC / RDST = CRHW
CRSEL1	J3 C21	slots 0–2 only	CPU	MMU/data	hardware control register for RSRC / RDST = CRHW
CRSEL2	J3 C22	slots 0–2 only	CPU	MMU/data	hardware control register for RSRC / RDST = CRHW
FAULT	J3 B4	slots 0–2 only	MMU/data	CPU	this memory cycle faulted: abort it, enter trap
FC0	J3 B6	slots 0–2 only	MMU/data	CPU	fault cause
FC1	J3 B8	slots 0–2 only	MMU/data	CPU	fault cause
FC2	J3 B10	slots 0–2 only	MMU/data	CPU	fault cause
FC3	J3 B12	slots 0–2 only	MMU/data	CPU	fault cause
-WSTB	J3 B14	slots 0–2 only	CPU	MMU/data	write strobe: gates every SRAM write
STALL	J3 B16	slots 0–2 only	MMU/data	CPU	hold the microcycle: the bus is not yet ours (arbitration) or the cycle is waiting (-WAIT)
LOCKREQ	J3 B18	slots 0–2 only	CPU	MMU/data	this access and the next are one atomic pair (-LOCK)
PCINC	J3 B22	slots 0–2 only	CPU	MMU/data	the PC's +4 and MAR <- PC: the MMU / data card keeps a shadow PC (it loads from R with the PC and counts with it), so the next fetch's address needs no R cycle
MARPC	J3 B24	slots 0–2 only	CPU	MMU/data	the PC's +4 and MAR <- PC: the MMU / data card keeps a shadow PC (it loads from R with the PC and counts with it), so the next fetch's address needs no R cycle
MARINC	J3 B26	slots 0–2 only	CPU	MMU/data	MAR += 4 (MAR is a counter): LI32's value word, then the next fetch (D59)
CCLK	J3 B20	slots 0–2 only	CPU	MMU/data	the CPU's clock: CLK's cycles in which the CPU commits (none while STALL, halted or between steps)
-CPUIN	J3 B28	slots 0–2 only	the CPU card: the pin tied to its ground	the debug card (slot 0)	low: a CPU card is in slot 1 (D102). While it is low the debug card drives nothing alone: no CLK, no -RES, none of its groups
-MMUIN	J3 B30	slots 0–2 only	the MMU / data card: the pin tied to its ground	the debug card (slot 0)	low: an MMU / data card is in slot 2 (D102). While it is low the debug card alone leaves that card's lines to it: D, the bus cycle's lines and the grants

The backplane

docs/backplane.md

Status: NETLIST DONE (2026-10-26, D68, accepted by Ken 2026-10-27), checked against the bus definition and against every card's netlist in every slot it may use; laid out and routed by its generator (D79, §4: 0 DRC violations, 0 unconnected). **Changed 2026-11-11 (D99, Ken): ten slots, 1.5 inches apart, every part and the power entry at the back, 14 mm taller than the cards at the top and at the bottom with the edge mounting holes there, the slots labelled.** Not ordered. The netlist is `generators/card_backplane.py`, the board `generators/p32backplane.py`; the bus itself, pin by pin, is bus.md (`generators/p32bus.py`).

30 sockets, 122 parts (`python3 generators/card_backplane.py --bom`).

1. What it holds

Ten positions, 1.5 inches (38.1 mm) apart (D99; eleven at 1 inch before, D49, D64): the debug slot 0, then slots 1–9. Each has three female DIN 41612 C 3 × 32 sockets, vertical, through-hole: **S*n*J1, S*n*J2, S*n*J3**, mating the card's J1–J3.

Front and back (D99). Slot 0's end is the **front**; the other end, past slot 9, is the **back**, and holds every part that is not a socket or its bypass, and both power entries. The one exception is the +5V LED with its resistor, at the front edge where it can be seen.

Taller than the cards (D99). A card's connector edge is 366.7 mm; the board is **394.7 mm**, 14 mm past the cards' ends at the top and at the bottom, as a standard 9U backplane is. The frame's rails bolt to those two strips, outside the cards, and the corner and edge mounting holes are in them. **Every card goes in with its components facing the front** (toward slot 0): the sockets are turned so (§4).

Slot	Card	Its J3
0	the debug card (D64)	the local link, a third drop
1	CPU	the local link
2	MMU / data, with the arbiter	the local link
3–9	any: memory, I/O, disk, video	power only

Seven general slots: four memory cards, I/O, disk and video fill them. The bus definition still has an eleventh position, slot 10, and the arbiter a request pair for it; this backplane has no such slot (the MMU / data card pulls an empty slot's `-BREQ` up, so it is never granted).

The nets are the bus definition's. Every pin of every socket is on `p32bus.busnet(connector, pin, slot)`, so the backplane cannot disagree with docs/bus.md or with the cards:

- **J1 and J2 are bused to every slot**: the cycle (D0–31, A2–31, BE0–3, `-MRD`, `-MWR`, `-ACK`, `-WAIT`) and control (CLK, `-CLK`, `-RES`, PWROK, `-LOCK`, `-IRQ0–7`, `-IACK`). **Their 19 spare pins too** (J2 A22–A30, C21–C30; A19–A21 became `-FAST`, XCLK and `-XCLKEN`, D74, D76): a later signal needs no new backplane (as P8X's rev D).
- **The radial request lines.** Each requester's own `-BREQ` / `-BGNT` pins (J2 A16 / A17) run to their own pins in slot 2 (the arbiter: J2 C3–C20) and in slot 0, so the debug card sees every request as the arbiter does: `-BREQ3`–`-BREQ9`, `-BGNT3`–`-BGNT9`, and slot 0's own pair, `-BREQ0` / `-BGNT0`, to slot 2 (slot 10's pair joins slots 0 and 2 only). In slots 1 and 3–9, J2 C3–C20 are not connected.

- **J3, the local link** (LR0–31, RSRC, RDST, MEM, SIZE, SIGNED, SUPER, TLBW, CRSEL, FAULT, FC, `-WSTB`, STALL, LOCKREQ, PCINC, MARPC, MARINC, CCLK), joins slots 0, 1 and 2 only. In slots 3–9 a card's J3 carries power and ground; its signal pins are not connected. **J3's eight spare pins (B28, B30, C25–C30) join slots 0, 1 and 2 as well** (D99): spare private lines between the debug, CPU and MMU / data cards, should one be needed after the backplane is made. **Two were needed before it was** (D102): B28 is now `-CPUIN` and B30 `-MMUIN`, which the CPU card and the MMU / data card each tie to ground so that the debug card can tell they are in. No copper changed, only the two nets' names. C25–C30, six pins, are still spare.
- **Power on every socket**: pins 1–2 of every row VCC, 31–32 GND, row B's odd pins GND, and J2 row B's even pins VCC as well (D57): 32 VCC and 60 GND pins per card.

2. The parts no card can hold

Ref	Part	Why
P1, P2 / P3, P4	M4 bolt and ring lug, in a plated hole stitched to its plane	the power entry: +5V and GND, two pairs, both at the back (D78, D99)
U1	TL7705ACP (DIP-8)	**PWROK**: the +5 V supervisor (D78)
D1, R1	LED, 1k	**+5V**: the supply is on
R2	1k	`-ACK` pull-up
R3	680	`-WAIT` pull-up
R4	1k	`-LOCK` pull-up
R5	10k	PWROK pull-up
R6	10k	`-FAST` pull-up: the CPU card's clock jumper, open = 3 MHz (D74)
RN1	2.2k × 8, bussed (SIP-9)	`-IRQ0` – `-IRQ7` pull-ups
RN2–RN13	10k × 8, bussed (SIP-9)	hold D0–31, A2–31, BE0–3, `-MRD`, `-MWR`, `-IACK`; each network on eight consecutive pins of one connector column, so it stands beside the socket rows on the board (D79; some resistors unused)
R7, C1 / R8, C2	100, 150 pF	AC termination of CLK / `-CLK` past slot 9
C3–C42	100 nF beside each socket; 10 µF a slot	bypass
C43, C44	1000 µF 16 V	bulk, one at each power entry
C45, C46, C47	100 nF, 10 µF, 100 nF	U1's REF, its delay (CT), its bypass
H1–H18	M3 hole, plated, on GND	the subrack (D99): the four corners (H1–H4); in the 14 mm strip above the cards, in line with the gaps between slots 1–2, 3–4, 5–6 and 7–8 and over the parts at the back (H5–H9); the same in the strip below the cards (H10–H14); four mid-board, in the gaps between the sockets (H15–H18)

2.1 Power: a dedicated 5 V supply (D78)

The supply is a **dedicated +5 V switching supply**, 30 A at least and **40 A recommended** (a known brand, e.g. Mean Well LRS-200-5): the machine's worst case with four memory cards is 30.3 A, and typical about 17 A (power-budget.md). It is switched on the mains side (a switch or a switched, fused inlet on the case); the backplane has no power switch or jumper.

- **The entry: two pairs of M4 bolts**, +5V and GND, both at the back (D99: P1 / P2 between the J1 and J2 rows, P3 / P4 between the J2 and J3 rows, past slot 9). Each is a plated 4.3 mm hole ringed with vias into its plane, an M4 bolt and nut through it and a ring lug on **10 AWG** wire under the head. Both pairs come from the same supply's terminals,

so each carries about half the current and no one point runs hot; one pair alone works for a light machine. A ring lug cannot pull out as a clamped bare wire can.

- **PWROK** comes from **U1, a TL7705A supply supervisor** (TI): its open-collector `RESET` output holds PWROK low until +5 V is above 4.55 V (SENSE on VCC), then for t = 1.3 × 10⁴ × CT (10 µF: about 130 ms), and pulls it low again whenever +5 V dips below the threshold. The CPU card holds reset until PWROK is high and sees it fall first when power fails, as it would have with an ATX supply's PWR_OK. `RESIN` is tied high; REF has the datasheet's 100 nF.
- **+12 V is not on the backplane.** The disk card's IDE hard drives take +12 V (and their +5 V) from the supply's own outputs if it has a 12 V rail, or from a second, 12 V supply; CompactFlash needs 5 V only.

2.2 The pull-ups: a rule, checked against the cards

Every bus line on J1 and J2 that no card drives with a totem-pole output gets a resistor to VCC here, and no other line does. The generator reads every card's netlist to find each line's drivers, and fails if a line breaks the rule either way:

- **The open-drain lines** (74HC05, 74LS145): they rise only through the resistor, so it is sized for speed. A card lets `-WAIT` go at mid-cycle (on `-CLK`); it must be high at the next CLK edge, half a clock later, less the card's GAL and gate delays and the master's setup: about 60 ns at 5 MHz, 130 ns at 3 MHz. The line is about 200 pF (eleven sockets, their cards' inputs, 30 cm of track); reaching HCT's 2.0 V takes 0.5 RC. `-WAIT` **680 Ω**: 70 ns, with 7.4 mA into a 74HC05 at about 0.45 V (HCT reads up to 0.8 V as low). `-ACK` and `-LOCK` have a whole clock: **1k**. `-IRQ0–7` are levels: **2.2k**, as P32's.
- `-FAST` (D74): a static level, the CPU card's clock jumper, which grounds it when fitted (the 4 MHz clock). **10k** holds it high when the jumper is open, or when no CPU card is fitted (the debug card alone).
- **The three-state lines**: D0–31 (driven only during a read or a write), A2–31, BE0–3, `-MRD`, `-MWR`, `-IACK` (driven by the master, so loose while mastership changes hands, or with the MMU / data card out). **10k** only holds them, so no CMOS input floats (the MMU / data card reads D through 74HCT253s on every cycle). The MMU / data card's own 10k on D0–7 (no vector reads `0xFF`) stay; with these, 5k.
- **PWROK**: 10k, for the supervisor's open-collector output (D78).
- **Not here:** the radial `-BREQ` lines (the MMU / data card pulls them up, so an empty slot asks for nothing, D62), and J3's lines (between the CPU and the MMU / data card, which drive them).

2.3 Termination

CLK and `-CLK` clock every card. The CPU card drives them from slot 1, near one end; at the other end, **past slot 9**, each line has **100 Ω in series with 150 pF to ground** (AC termination, as P32's terminator card, D56 there): it damps the reflection from the open end while an edge passes, and draws no current in the steady state, so it is no load on a 74HCT or GAL driver. Values to settle on the bench (scope CLK at slots 0, 1 and 10, with every card in). The other lines change while the clock is steady, and are taken at its edges.

2.4 Bypass

A 100 nF (0.1" leads, D47) beside each of the 30 sockets, a 10 µF a slot, and two 1000 µF at the power entry. Each card has its own bypass and bulk capacitors as well.

3. Verification

`python3 generators/card_backplane.py` (in `make test`):

- the netlist's electrical checks (p32net: every pin connected or declared unused, no net with one end);
- every pin of every socket against `busnet()`; every bus net reaches at least two sockets;
- **every card in every slot it may use**: the CPU card in slot 1, the MMU / data card in 2, the debug card in 0, the memory, I/O, disk and video cards in 3–9: each connector pin the card uses must be wired there (a device card using a J3 signal, for instance, would fail);
- the pull-up rule above, against every card's drivers; termination only on lines a card drives totem-pole.

The gate-level system simulation (`generators/sim_sys.py`) joins the cards by these same net names, with the open-drain lines' pull-ups modelled.

4. The board (D79)

`generators/p32backplane.py` places **and routes** it; no autorouter. `sh generators/build_board.sh backplane` adds the gerbers, renders, the placement drawing and the DRC.

- **477 × 394.7 mm, 6 layers** (D77, D90, D99): F.Cu, In1 GND, In2, In3, In4 VCC, B.Cu. Slot 0 at the left (the front), 38.1 mm apart; 34 mm of board in front of slot 0, the room its card's components have; 14 mm above and below the cards' ends (a card's top edge is 14 mm below the board's).
- **The sockets** are turned 180°, so that pin 1 is at the bottom as on the cards: each socket's a1 height comes from the card generator's own placement of J1–J3, so the two cannot disagree. Turned so, column c is on the front side of column a. On a card's right-angle connector row a is the row nearest its board and row c the farthest above it, so **a card's components face the front** (D99). That reading is from KiCad's footprints of the two connectors: **VERIFY with a real connector and socket before ordering.**
- **Routing:** a bused net is the same pin in every slot, at one height. Each pin drops a 1.27 mm stub into the channel between its row and the next, and one straight track runs along the channel: column a on F.Cu, b on In2, c on In3. The request lines (`-BREQn` / `-BGNTn`): the C pins in slots 0 and 2 are one channel, carried on along the same row to slot n; there a via and a riser on B.Cu, between two pin columns, climb to the slot's A16 or A17. Every track 0.2 mm, at least 0.395 mm from the next pad.
- **Everything else is at the back** (D99), in a field past slot 9: the pull-ups, every network pin and resistor end on its own net's row, so each net's channel simply runs on to it; the termination (R7 / C1, R8 / C2) and the **TL7705A**, on PWROK's row, among them.
- **Power:** the two pairs of M4 bolts (+5V above GND) at the back, in the gaps between the socket rows where no channel runs, a 1000 µF beside each pair; the **+5V LED (D1) and its resistor (R1) at the front edge**, 6 mm in, at the J1 end; the bypass capacitors in the gaps beside each slot; the planes reach every VCC and GND pin.
- **Mounting holes** (D99): 18, M3, plated and on GND. Fourteen are in the two 14 mm strips past the cards' ends, 7 mm from the board's edge: the corners, one in line with every second gap between slots (1–2, 3–4, 5–6 and 7–8: 76.2 mm apart) and one over the field. Four are mid-board, in the same four gaps, between the socket rows.
- **Silkscreen:** along the front edge, reading from the front: FRONT, **the board's name, revision and `cottageworker.com`, 7 mm tall**, and "CARDS GO IN WITH THEIR COMPONENTS FACING THE FRONT (SLOT 0)"; BACK along the back edge; **each slot's number and what it is for, 3 mm and 2.5 mm tall**, between its J1 and J2 sockets: SLOT 0 DEBUG, SLOT 1 CPU, SLOT 2 MMU/DATA, SLOT 3–9 GENERAL PURPOSE (D99); **"a1" beside pin a1 of every socket**, outside the socket's body (below); +5V and GND at each bolt, +5V at the LED, every reference beside its part.
- **Fitting the sockets.** The board's wiring fixes how a *card* must stand: its components toward the front and its pin 1 at the bottom, so that its pin a1 meets the hole wired as a1 (on the memory card J1's a1 is 89.37 mm below the card's

top edge, and so is the board's a1 hole; J2 222.72, J3 356.07). The socket itself is 96 straight-through contacts, the same electrically either way round, and its markings are only markings (Ken); but its keyed shell decides which way a card can be pushed in. So each socket is soldered the way round that takes a card standing as above: by the connectors' own markings that is with the socket's a1 at the board's printed "a1". The other way round, a card would only go in turned about, with its a1 in the hole wired as c32: every signal on the wrong pin and +5 V and ground swapped (pins 1–2 are +5 V, 31–32 ground). To try with a real card connector before soldering. A socket's two fixing holes are 0.3 mm off its pins' centre line, so turned about they sit 0.6 mm from the board's.

- **Renders:** top, bottom and 3D (`backplane-render-*.png`), as for every board since 2026-11-11.
- **The supply:** a dedicated +5 V supply, 40 A recommended (§2.1, D78; docs/power-budget.md).

The backplane: theory of operation

docs/backplane-theory.md

Status: DESCRIBES THE NETLIST AND THE BOARD AS GENERATED AND CHECKED (2026-10-29): the netlist `generators/card_backplane.py` and the board `generators/p32backplane.py`, on the bus definition `generators/p32bus.py`. The board's summary and parts list are in the backplane; the bus, pin by pin, in the bus. D49, D50, D52, D57, D64, D68, D74, D76, D77, D78, D79. **Brought up to D99 (2026-11-11, Ken): ten slots, 1.5 inches apart, front and back, every part and the power at the back, 18 mounting holes, the slots labelled.**

1. What the backplane does

The backplane is **passive**: sockets, copper and the parts no card can hold (the pull-ups, the clock's termination, the power entry, a +5V LED, the capacitors, the mounting holes, and its only chip, U1, the supply supervisor that makes `PWROK`). It carries **the bus** (D50: physical memory cycles, interrupts, arbitration, the clock, reset and power) to ten positions, and **the local link** (J3) between the CPU card and the MMU / data card. Every socket pin's net **is** `p32bus.busnet(connector, pin, slot)`, from the file the cards take their pins from, so the backplane cannot disagree with the bus or the cards: **30 sockets, 122 parts, 194 bus nets** (`card_backplane.py --bom`).

2. The slots

Ten positions, **1.5 inches (38.1 mm) apart** (D99; eleven at 1 inch before, D49, D64), slot 0 at the left: the **front**. The other end, past slot 9, is the **back**. Each position has three female DIN 41612 C 3 × 32 sockets, S*n*J1–S*n*J3, mating a card's J1–J3. **Every card goes in with its components facing the front**: on the board a socket's column c is on the front side of its column a, and on a card's right-angle connector row c is the row farthest above the card (§10; to verify with real parts).

Slot	Card	Its J3	Silkscreen
0	the debug card (D64): listens to every signal, or drives, never both	the local link, a third drop	SLOT 0, DEBUG
1	the CPU card: drives `CLK`, `-CLK`, `-RES`; its CLOCK jumper, fitted, grounds `-FAST`	the local link	SLOT 1, CPU
2	the MMU / data card: the CPU's bus master, and the arbiter	the local link	SLOT 2, MMU/DATA
3–9	any: memory, I/O, disk, video	power only	SLOT *n*, GENERAL PURPOSE

J3, the local link (62 lines: R as `LR0–31`, the microword's memory fields, `FAULT`, `FC0–3`, `STALL`, `CCLK`, …; nets `L_LR0`, `L_STALL`, …) joins slots 0, 1 and 2 only, so the microcode's memory controls never become bus signals (D49). **Slots 3–9** take any card: a card is placed by its jumpers, not its slot (D50, D52), and the slot sets only its bus-request priority. The bus definition has a slot 10 as well, and the arbiter a request pair for it; this backplane has no such position (D99).

3. Bused lines and per-slot lines

Power on every socket: pins 1–2 of every row VCC, 31–32 GND, row B's odd pins GND (a guard between rows A and C), J2 row B's even pins VCC too (D57): **32 VCC and 60 GND pins per card**.

Bused to all ten slots, the same pin in every slot: **J1, the cycle** (70 lines: `D0–31`, `A2–31`, `BE0–3`, `-MRD`, `-MWR`, `-ACK`, `-WAIT`; J1 has no spare pins); **J2, control** (17 lines: `CLK`, `-CLK`, `-RES`, `PWROK`, `-LOCK`, `-IRQ0–7` on A3–A15, and `-IACK`, `-FAST`, `XCLK`, `-XCLKEN` on A18–A21); and **J2's 19 spare pins** (A22–A30, C21–C30, D68), so a later signal needs no new backplane.

Per slot: the radial request lines. A requester's own `-BREQ` / `-BGNT` pins are J2 A16 / A17, and each slot's pair is two nets of its own:

Nets	From	To
`-BREQn`, `-BGNTn` (n = 3–9)	slot *n*'s J2 A16, A17	the same J2 C pin (C3–C18) in slot 2, the arbiter, and in slot 0, the debug card
`-BREQ10`, `-BGNT10`	no slot (D99)	their J2 C pins in slots 2 and 0 only
`-BREQ0`, `-BGNT0`	slot 0's J2 A16, A17	slot 2's J2 C19, C20

Not connected: J2 A16–A17 in slots 1 and 2 (neither card asks for the bus), J2 C3–C20 in slots 1 and 3–9, J2 C19–C20 in slot 0, J3's signal pins in slots 3–9, and J3's spare pins there too. In slots 0, 1 and 2 J3's pins B28, B30 and C25–C30, spare when the backplane was drawn, are joined, slot to slot, as the local link is (D99). Two of the eight are now signals (D102): B28 is `-CPUIN` and B30 `-MMUIN` (nets `L_-CPUIN`, `L_-MMUIN`), each tied to ground by the card it names and pulled up on the debug card, which will not drive a line of a card that is in (debug-card-theory §9). The backplane only joins them: it has no part on either. C25–C30 (nets `L_SPARE_J3C25`, …) are six spare private lines.

4. Requests, grants and the lock

The backplane only carries the radial lines; the arbiter is **GAL ARB** on the MMU / data card, clocked by `CLK` (mmu-card-theory §9). At a `CLK` edge with no grant out and `-LOCK` high, the requesting slot of the highest priority, **slot 0 first, then the highest-numbered slot**, gets its grant and is master **from the next clock**. It keeps the bus as long as it holds its `-BREQ` low; `-RES` clears every grant. GAL ARB drives the `-BGNT` lines totem-pole. The disk card's DMA engine (D75) asks from GAL DKENG, the debug card through a 74HC05. **The request lines' pull-ups are on the MMU / data card**, not here: a 10k on each of `-BREQ0` and `-BREQ3–10` holds an empty slot's request high.

`-LOCK` (J2 A7) is open-drain, from a 74HC05 on the MMU / data card (a `SWAP`'s read and write, D51, D54) and one on the debug card; while it is low the arbiter grants nothing. **The masters** (the MMU / data card for the CPU, the disk card's DMA engine, the debug card) drive the address, byte enables, `-MRD` / `-MWR` and a write's data through three-state drivers; only the MMU / data card and the debug card drive `-IACK`.

5. The pull-ups

The rule (D68): every J1 / J2 bus line that **no card drives with a totem-pole output** gets a resistor to VCC here, and no other line does. The generator reads every card's drivers and fails if a line breaks the rule either way.

Line	Pin	Ref	Value	Driven by
`-ACK`	J1 B28	R2	1k	74HC05, open drain: the memory, I/O, disk, video, debug cards
`-WAIT`	J1 B30	R3	680 Ω	74HC05: the I/O, disk, video, debug cards
`-LOCK`	J2 A7	R4	1k	74HC05: the MMU / data, debug cards
`PWROK`	J2 A6	R5	10k	U1, open collector (§8.2)
`-FAST`	J2 A19	R6	10k	the CPU card's CLOCK jumper: fitted, ground (4 MHz); open, or no CPU card, high (3 MHz)
`-IRQ0–7`	J2 A8–A15	RN1	2.2k × 8	74LS145, open collector: the I/O (nine), disk, video cards

Line	Pin	Ref	Value	Driven by
`D0–31`, `A2–31`, `BE0–3`, `-MRD`, `-MWR`	J1	RN2–RN12	10k × 8	three-state: the masters; on a read, the addressed card's `D`
`-IACK`	J2 A18	RN13	10k × 8	three-state: the MMU / data, debug cards

- **The open-drain lines** rise only through their resistor, so it is sized for speed: about 200 pF of line (ten sockets, the cards' inputs, 45 cm of track) and 0.5 RC to reach HCT's 2.0 V (backplane.md §2.2). `-WAIT` is let go at mid-cycle (on `-CLK`) and must be high by the next `CLK` edge: 680 Ω, about 70 ns, 7.4 mA into a 74HC05 at about 0.45 V. `-ACK` and `-LOCK` have a whole clock: 1k. `-IRQ0–7` are levels: 2.2k.
- `PWROK` (U1's open-collector output, D78) and `-FAST` (a static level, D74): 10k.
- **The three-state lines** are loose whenever no one drives them: `D` outside a read or a write; the address, byte enables and strobes while mastership changes hands, or with the MMU / data card out. 10k only holds them, so no CMOS input floats (the MMU / data card reads `D` through 74HCT253s on every cycle); a held strobe is high, no cycle. A device master drives no `-IACK` (D75), so RN13 holds it off during a disk DMA.
- **No resistor** on the lines a card drives totem-pole: `CLK`, `-CLK`, `-RES` (the CPU card's GALs CLKGEN and PANEL), `XCLK`, `-XCLKEN` (the debug card's GAL DBCLK); none on the request lines (§4) or on J3.

The networks are bussed SIP-9s, each on eight consecutive pins of one connector column (`pull_networks()`, D79), so that on the board every pin stands on its own net's row: RN2–RN5, RN6–RN8, RN9–RN12 on J1 columns A, B, C; RN13 on J2 A18–A25. A row with no such net leaves its resistor unused: 27 of their 96, half of RN6–RN8 (row B's odd pins are GND), seven of RN13's. They stand, with R2–R6 and the termination, in a field past slot 9, at the back, where each net's track ends.

6. The clock

`CLK` **and** `-CLK` (J2 A3, A4) come from the CPU card in slot 1, from GAL CLKGEN; `CLK` is high in phases 0–3 of the eight (cpu-card-theory §3). One period is one bus cycle: 333 ns at 3 MHz, or 4 MHz with the CLOCK jumper (D74). Every card works from `CLK`. Each line is one straight track past all ten slots.

The termination is at the far end from the CPU card, **past slot 9**: on each line **100 Ω in series with 150 pF to ground** (R7 / C1 on `CLK`, R8 / C2 on `-CLK`). It damps the reflection from the open end while an edge passes and draws no current in the steady state, so it is no load on the GAL driving the line; the values are to settle on the bench (backplane.md §2.3). The other lines are not terminated: they change while the clock is steady and are taken at its edges.

The clock's other lines, on J2: with `-FAST` (A19) low the I/O card's timer divides by 4, so it ticks at 1 MHz at either clock (D74). `XCLK` and `-XCLKEN` (A20, A21; D76) come from the debug card's GAL DBCLK; the CPU card's GAL CKSEL takes `XCLK` as its oscillator while `-XCLKEN` is low (12 MHz, or one edge per host STEP). With no debug card, the CPU card's own 10k resistors hold `XCLK` low and `-XCLKEN` high. With no CPU card, the debug card makes `CLK` and `-CLK` itself (GAL DBCLK) and can drive `-RES`, through three-state drivers (D64).

7. Interrupts and the acknowledge cycle

`-IRQ0–7` (J2 A8–A15) are eight bused open-drain levels, held high by RN1. A device's level is set by jumpers on its card (D52) and decoded onto its line by a 74LS145, open collector: one per device on the I/O card, one on the disk card, one on the video card. The CPU card takes the levels through a 74ACT574 at `-CLK`'s rise (`card_cpu.py`); the debug card listens. No level depends on a slot. A level belongs to **one card**; one card may drive several (D57).

The acknowledge cycle (D57) is a bus cycle like a read, with `-IACK` (J2 A18) low in place of `-MRD` and `A4:A2` = the level; the MMU / data card runs it for the CPU. The card driving that level pulls `-ACK` and puts its vector, window × 16 + the asking device, on `D7:D0`. **No answer is not a bus error:** R2 keeps `-ACK` high, and the vector reads `0xFF` through RN2 and the MMU / data card's own 10k on `D0–7` (5k together). The kernel then reads the cards' interrupt status registers.

8. Power

8.1 The supply, the bolts and the capacitors (D78)

A dedicated +5 V switching supply, 30 A at least, **40 A recommended** (e.g. Mean Well LRS-200-5): the machine with four memory cards is 16.9 A typical and 30.3 A at its datasheet worst case, the backplane itself 5 mA, 79 mA at most (power budget). It is switched on the mains side; the backplane has no power switch or jumper. **+12 V is not on the backplane:** the IDE drives take it from the supply's 12 V rail or a second supply, and their +5 V from the supply directly.

The entry is two pairs of M4 bolts, both at the back (D99): P1 (+5V) and P2 (GND) in the gap between the J1 and J2 rows, P3 (+5V) and P4 (GND) in the gap between the J2 and J3 rows, over the field past slot 9, where no channel runs: a plated 4.3 mm hole ringed with vias into its plane, a ring lug on 10 AWG wire under the bolt's head. Both pairs come from the same supply, so each carries about half the current. **The planes**, In1 GND and In4 VCC, reach every socket's VCC and GND pins; a card's 32 VCC and 60 GND contacts are rated 1–2 A each, against the largest card's 7.8 A at most. **H1–H18**, M3 plated holes for the subrack, are on GND (D99). The board is 14 mm taller than the cards at the top and at the bottom (394.7 mm against 366.7), and fourteen of the holes are in those two strips, outside the cards' ends: the corners, one in line with every second gap between slots (1–2, 3–4, 5–6, 7–8: 76.2 mm apart) and one over the field. Four more are mid-board, in the gaps between the socket rows.

The capacitors: a 100 nF beside each of the 30 sockets and a 10 µF a slot (C3–C42, in the gaps between the J1, J2 and J3 sockets), and a 1000 µF 16 V beside each pair of bolts (C43 by P1 / P2, C44 by P3 / P4); every card has its own as well. **D1**, an LED through R1 (1k), labelled +5V, shows the supply is on: the two stand at the front edge, where they can be seen (D99), the only parts there.

8.2 PWROK: the supervisor (D78)

U1, a TL7705A (TL7705ACP, DIP-8), stands at the back with the other parts (D99), on `PWROK`'s row, so the line runs through every slot and on to R5 and to it:

Pin	Net	Use
8 VCC, 7 SENSE	VCC	it senses +5 V itself: threshold 4.55 V
2 `-RESIN`	VCC	tied high
1 REF	C45, 100 nF to GND	the datasheet's reference capacitor
3 CT	C46, 10 µF to GND	the delay, 1.3 × 10⁴ × CT: about 130 ms
5 `-RESET`	`PWROK`	open collector, pulled up by R5 (10k)
6 `RESET`	—	unused

`PWROK` is low until +5 V is above 4.55 V, then for about 130 ms more, and low again whenever +5 V dips below the threshold. The CPU card's GAL PANEL registers `RESET = RESBTN or not PWROK` and drives `-RES` onto the bus (boot §1), so every card is held in reset until power is good. The debug card listens to `PWROK`. C51 (100 nF) decouples U1.

9. An empty slot

Nothing need be fitted in any slot. An empty socket stays on every bused net, and the bus sees **no** `-ACK` at the addresses its card would answer: R2 holds `-ACK` high and the MMU / data card makes the cycle a **bus error**, so a missing card never hangs the bus (the kernel finds the cards this way, D52). It sees **no wait** (R3), **no interrupt** (RN1; an acknowledge cycle on a level no card drives reads `0xFF`), and **no request**: the MMU / data card's 10k holds the slot's `-BREQn` high, so GAL ARB never grants it. In slots 3–9, J3 is power only. Three are special:

- **Slot 0 empty** (no debug card): the MMU / data card's 10k holds `-BREQ0` high; the CPU card's resistors hold `XCLK` low and `-XCLKEN` high, so the CPU card keeps its own oscillator.
- **Slot 1 empty** (the debug card alone, D64): R6 holds `-FAST` high, the 3 MHz setting; `CLK`, `-CLK` and `-RES` have no driver but the debug card.
- **Slot 2 empty:** no arbiter, and no pull-ups on the request lines; the backplane's 10k still holds `D`, the address, byte enables and strobes.

10. Verified

`python3 generators/card_backplane.py`, in `make test`, checks the electrical rules (`p32net.check`: every pin connected or declared unused, no pin on two nets, no net with one end, at most one totem-pole driver); every pin of every socket in every slot against `busnet()`, every bus net and spare on two sockets or more; **every card in every slot it may use** (debug 0, CPU 1, MMU / data 2, memory, I/O, disk, video 3–9), each pin the card uses wired there; the pull-up rule against every card's drivers, both ways; and termination only on lines a card drives totem-pole. It prints `backplane OK: 10 slots (0-9), 30 sockets, 194 bus nets, 122 parts; every card fits its slots`. `p32bus.py --check`, also in `make test` (which keeps docs/bus.md current), checks each signal's one pin, the local link's `RSRC`, `RDST` and `MEM` widths against the microword, and that each radial line reaches its own slot, slot 2 and, from slots 3–10, slot 0.

The board (D77, D79): `p32backplane.py` places **and routes** it, with no autorouter: **477 × 394.7 mm, 6 layers** (F.Cu, In1 GND, In2, In3, In4 VCC, B.Cu; D99), the sockets turned 180° so that pin 1 is at the bottom as on the cards and column c is on the front side of column a (so a card's components face the front, by KiCad's footprints of the two connectors), each bused net one straight track. It asserts each socket's pins where the cards put theirs, each network on eight rows of one column, and no two nets' tracks overlapping on a layer; KiCad's DRC found **0 violations, 0 unconnected**. **Not yet checked:** the termination's values (bench), and the sockets' orientation against a real card (backplane.md §4).

11. Open questions

1. **Interrupt priority.** Settled 2026-11-11: bus.md's signal table said `-IRQn` "(IRQ0 = timer; 7 highest)"; it now says what its Interrupts text, isa-system.md, cpu-card-theory §14 and the emulators do: the lowest-numbered level first.
2. **The clock rates.** bus.md's cycle text said "5 MHz planned, 3 MHz as the safe setting"; since 2026-11-11 it says 3 MHz, or 4 MHz with the jumper (D72, D74). Still open: backplane.md §2.2 budgets `-WAIT` at 5 and 3 MHz only, and its 680 Ω rise (about 70 ns) is over its own 5 MHz budget (about 60 ns).
3. **When the arbiter grants.** bus.md: "when the CPU's next microcycle does not use the bus"; GAL ARB has no such input and grants at any `CLK` edge with no grant out and `-LOCK` high (mmu-card-theory §9). What becomes of a CPU cycle stretched by `-WAIT` when a grant falls on one of its clock edges is not said.

4. `XCLK` **is not terminated**, though it is a clock (12 MHz, or single edges, D76) driven from slot 0 into the CPU card's oscillator select; backplane.md §2.3's reason for terminating only `CLK` and `-CLK` does not cover it.
5. **Slot 2 empty:** `-BGNT0` and `-BGNT3–10` are driven, and `-BREQ0` and `-BREQ3–10` held, only by the MMU / data card; with the debug card alone nothing drives the grants or holds the requests. Intended?
6. **Stale comments and names.** (The mounting holes: settled by D99, which puts them at the corners, along both edges and between the sockets, in the netlist, the board and backplane.md alike.) p32bus.py's comments give J3 as "slots 1-2" and the "arb" lines as "slot 2 only"; `busnet()` wires both to slot 0 too. D78 and backplane.md §2.1 call U1's pins 5 and 2 `RESET` and `RESIN`, the netlist `-RESET` and `-RESIN` (pin 6, `RESET`, is unused). cpu-card-theory §14 (a 74HCT574 on `-IRQ0–7`) and mmu-card-theory §6 (74HCT541s on the address) predate D72's 74ACT574 and 74ACT541s.