Sonne-8 Microcontroller Reference Schematics Rev. Myth

PHA0: Put page index (C) and byte offset (PC) of instruction on address bus
PHA1: Latch opcode or IRQ Trap opcode into IB0-7, pre-decode instruction
PHA2: Decode instruction, set up source byte on DB or other action
PHA3: Latch results into target registers or other action
CLOSE: Finish latching etc.

Prevent same-cycle memory-to-memory copy. Example: MxM and FxM will not execute

Instruction byte:

```
all0: OPC_SYS       00000   xxx    See table @ SYS decoder
else: OPC_STASH     00001   xxx    See table @ STASH decoder
      OPC_ALU       0001   xxxx    See table @ ALU
      OPC_TRAP      001   xxxxx    b0-4: DEST PAGE
      OPC_GETPUT    01 xx x xxx    b0-2 OFFS, b3:GET/PUT, B4-5:REG(GETPUT)
      OPC_PAIR      1  xxx xxxx    b0-3 DST, b4-6 SRC
```

Start, latch instruction or interrupt vector

Interrupts disabled when "BUSY" and when running code in page 0 (See code page index reg schematics bottom right)

On instruction fetch: If IRQ and not BUSY, inject "TRAP 0" opcode, else fetch opcode from memory at C:PC

TRAP opcode triggers call to interrupt service routine in page 0, disabling further interrupts, and toggles ("BUSY")

BUSY state removed by RTI instruction

Hardwire IRQ TRAP instruction: 001.00000 to page 0

ALU Instructions

0 IDA ("Identity A")	A := A
1 IDT ("Identity T")	A := T
2 OCA ("One's Complement T")	A := ~A
3 OCT ("One's Complement T")	A := ~T
4 SLA ("A shifted left")	A := A << 1
5 SLT ("T shifted left")	A := T << 1
6 SRA ("A shifted right")	A := A >> 1
7 SRT ("T shifted right")	A := T >> 1
8 AND ("A boolean-and T")	A := A & T
9 IOR ("A inclusive-or T")	A := A | T
10 EOR ("A exclusive-or T")	A := A ^ T
11 ADD ("Add T to A)	A := A + T, bits 0-7
12 CAR ("Carry bit of: A plus T")	A := 9th bit of: A + T, 0 or 1
13 AL_ ("Is A less than T")	A := (A<T)? 0xFF:0
14 AE_ ("Is A equal to T")	A := (A==T)? 0xFF:0
15 AG_ ("Is A greater than T")	A := (A>T)? 0xFF:0

O:
- Saves offset of current instruction byte during call/trap/go
- Outputs return offset during RET
- Is the (only!) implied address offset for memory read-write, with the corresponding page index stored in B (Base)
- Together with B forms 16-bit address pointers subject to the xU instruction.
  The B:O pointer can be loaded from/saved into the BO-Pointers P1-4

B:
- Saves page index of current instruction byte during call/trap/go
- Outputs return page index during RET
- Is the implied page index for memory read-write, with the corresponding byte offset stored in O (Offset)
- Together with O forms a 16-bit address pointers subject to the xU instruction
  The B:O pointer can be loaded from/saved into the BO-Pointers P1-4

Enable memory writes:
-Only during latch cycle of data phase (PHA3)
-When writing a memory value (xM),
 if this instruction is not conflicting (scrounge)
-During PUT type GETPUT instructions (writing to local memory)

Enable memory output:
-During IPH (instruction fetch phases)
-When fetching a literal (Fx) or memory value (Mx),
 if these are not conflicting (scrounge)
-During GET type GETPUT instructions (reading from local memory)

Based on working prototype PCB: https://github.com/michaelmangelsdorf/sonne8
Rev. Step "ToV"
Project homepage: https://github.com/michaelmangelsdorf/Myth
Author: Michael Mangelsdorf
Contact: mim@ok-schalter.de

Title: Sonne-8 Micro-Controller

Date: 2025-06-04

Rev: Myth