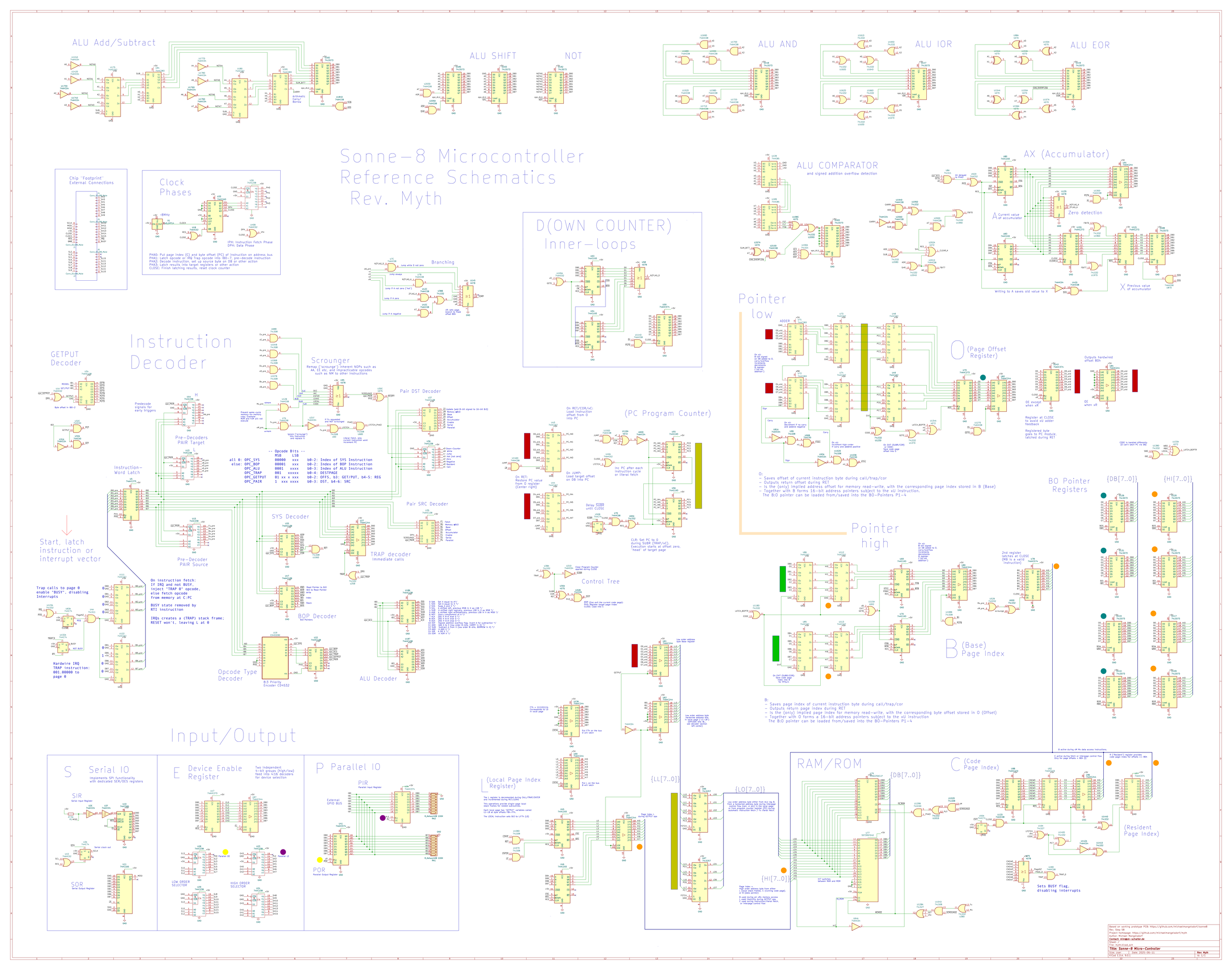
Sonne-8 Microcontroller
Reference Schematics
Rev. Myth
ALU Add/Subtract
ALU SHIFT
NOT
ALU AND
ALU IOR
ALU EOR
ALU COMPARATOR
and signed addition overflow detection
AX (Accumulator)
Chip "Footprint"
External Connections
Clock
Phases
~8MHz
IPH: Instruction Fetch Phase
DPH: Data Phase
PHA0: Put page index (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 results, reset clock counter
D(OWN COUNTER)
Inner-loops
Branching
Jump while D not zero
Jump always
Jump if A not zero ("not")
Jump if A zero
Jump if A negative
xR: sets page, selects 16 fixed offset BOH
Pointer
low
ADDER
(Page Offset
Register)
Outputs hardwired
offset BOh
OE except
when xR
Register at CLOSE
to avoid xU adder
feedback
Registered byte
goes to PC module,
latched during RET
OE
when xR
CDE is handled differently
(D can't latch PC via DB)
O:
- Saves offset of current instruction byte during call/trap/cor
- 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
Instruction
Decoder
GETPUT
Decoder
Byte offset in IB0-2
Predecode
signals for
early triggers
Pre-Decoders
PAIR Target
Instruction-
Word Latch
Start, latch
instruction or
interrupt vector
Pre-Decoder
PAIR Source
Scrounger
Remap ("scrounge") inherent NOPs such as
AA, EE etc, and impracticable opcodes
such as NM to other instructions
Prevent same-cycle
memory-to-memory
copy. Example:
xR,xR,xR fall will not
work
Ignore ("scrounge")
this instruction
and replace it
Literal fetch, skip
current instruction word
increment PC
Pair DST Decoder
Pair SRC Decoder
SYS Decoder
TRAP decoder
Immediate calls
BOP Decoder
BO Pointers
Read Pointer to BO
BO to Read Pointer
Opcode Type
Decoder
8:3 Priority
Encoder CD4532
ALU Decoder
-- Opcode Bits --
MSB LSB
all 0: OPC_SYS 00000 xxx D0-2: Index of SYS Instruction
else: OPC_BOP 00001 xxx D0-2: Index of BOP Instruction
OPC_ALU 0001 xxxx D0-3: Index of ALU Instruction
OPC_TRAP 001 xxxxx D0-4: DESTPAGE
OPC_GETPUT 01 xx x xxx D0-2: OFFS, D3: GET/PUT, D4-5: REG
OPC_PAIR 1 xxx xxxx D0-3: DST, D4-6: SRC
On instruction fetch:
If IRQ and not BUSY,
inject "TRAP 0" opcode,
else fetch opcode
from memory at C:PC
BUSY state removed by
RTI instruction
IRQs creates a (TRAP) stack frame;
RESET won't, leaving L at 0
Trap calls to page 0
enable "BUSY", disabling
Interrupts
Hardwire IRQ
TRAP instruction:
001.00000 to
page 0
On RET:
Restore PC value
from O register
(Center right)
On JUMP:
Load target offset
on DB into PC
On RET/COR/xC:
Load instruction
offset from O
into PC
(PC Program Counter)
Inc PC after each
instruction cycle
or literal fetch
Delay SUBR
until CLOSE
CLR: Set PC to 0
during SUBR (TRAP/xC).
Execution starts at offset zero,
"head" of target page
Control Tree
Clear Program Counter
replied during CLOSE
EXT (Can exit the current code page):
Set page index to current page-index
CLOSE: Load into B
BO Pointer
Registers
{DB[7..0]}
{HI[7..0]}
Pointer
high
2nd register
latches at CLOSE
(MB is a valid
instruction)
B (Base)
Page Index
B:
- Saves page index of current instruction byte during call/trap/cor
- Outputs return page index during RET
- Is the (only) 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
Input/Output
S Serial IO
Implements SPI functionality
with dedicated SER/DES registers
SIR
Serial Input Register
Serial clock out
SOR
Serial Output Register
E Device Enable
Register
Two independent
4-bit groups (high/low)
feed into 4:16 decoders
for device selection
LOW ORDER
SELECTOR
HIGH ORDER
SELECTOR
P Parallel IO
PIR
Parallel Input Register
External
GPIO BUS
POR
Parallel Output Register
L (Local Page Index
Register)
The L register is decremented during CALL/TRAP/ENTER
and incremented during RET/CLOSE
This operations provide single-page local
stack frames for nested subroutines.
Each local page has "GETPUT" variables called
L5-L8 at byte offsets F8h-FFh.
The LOCAL instruction sets B:O to L7:h (L0)
{LL[7..0]}
{LO[7..0]}
{HI[7..0]}
Low order address byte
taken from ALU reg B,
from a hardwired address byte during immediate
fetch, from D (data pointer), or from O
or from program counter register (PC) during
automatic instruction fetch ("7.fx literal fetch.
Page index =
high order address byte from either:
L (local stack frame); C (running code page),
or D (data pointer)
B used during all xMx memory access
O used during automatic PC fetch,
L used during instruction/literal fetch,
or interpage control flow
RAM/ROM
C (Code
Page Index)
(Resident
Page Index)
Sets BUSY flag,
disabling interrupts
Based on working prototype PCB: https://github.com/michaelmangelsdorf/sonne8
Rev. Step 58
Project homepage: https://github.com/michaelmangelsdorf/myth
Author: Michael Mangelsdorf
Contact: mimg@sk-schalter.de
Sheet: /
File: myth.kicad_sch
Title: Sonne-8 Micro-Controller
Size: User Date: 2025-06-11 Rev: Myth
KiCad 8.0.4 9.0.1 Id: 1/1