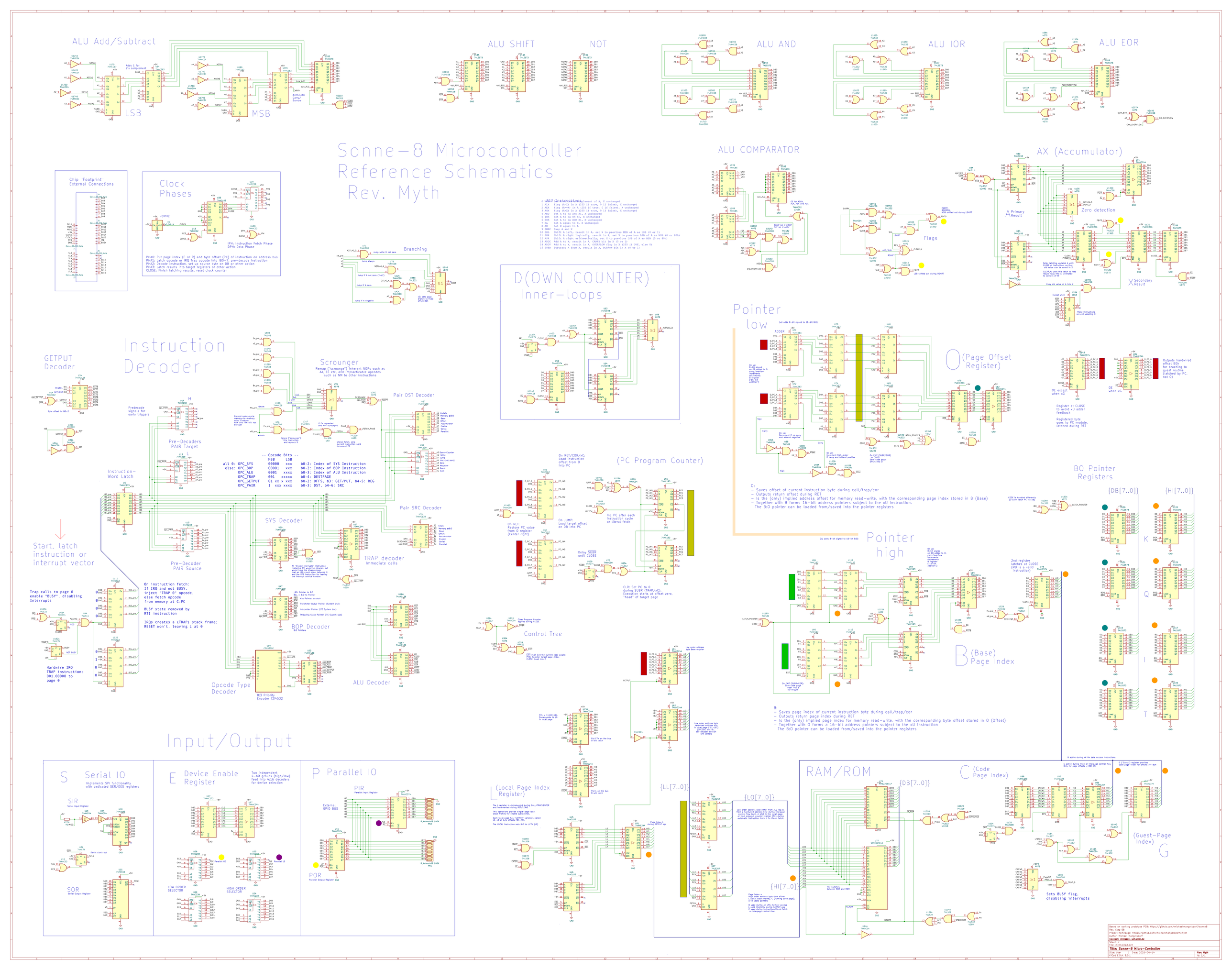
Sonne-8 Microcontroller
Reference Schematics
Rev. Myth
ALU Add/Subtract
LSB
MSB
Adds 1 for 2's complement
Arithmetic Carry/Borrow
ALU SHIFT
NOT
ALU AND
ALU IOR
ALU EOR
ALU COMPARATOR
Flags
AX (Accumulator)
Zero detection
Primary Result
Secondary Result
Chip "Footprint" External Connections
Clock Phases
8MHz
IPH: Instruction Fetch Phase
DPH: Data Phase
PHA0: Put page index (C or R) and byte offset (PC) of instruction on address bus
PHA1: Latch opcode or RQ 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
Branching
Jump always
Jump if A not zero ("NZ")
Jump if A zero
Jump if A negative
Jump while D not zero
xG sets page, branch to fixed offset B0h
D(OWN COUNTER)
Inner-loops
Pointer low
(Page Offset Register)
Instruction Decoder
GETPUT Decoder
Byte offset in IB0-2
Scrounger
Remap ("scrounge") inherent NOPs such as AA, EE etc. and impracticable opcodes such as NM to other instructions
Pair DST Decoder
Pair SRC Decoder
Predecode signals for early triggers
Pre-Decoders PAIR Target
Pre-Decoder PAIR Source
Instruction-Word Latch
Start, latch instruction or interrupt vector
SYS Decoder
TRAP decoder
Immediate calls
BOP Decoder
BO Pointers
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
Trap calls to page 0 enable "BUSY", disabling Interrupts
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
Hardwire IRQ TRAP instruction: 001.00000 to page 0
Pointer high
BO Pointer Registers
{DB[7..0]}
{HI[7..0]}
Control Tree
(PC Program Counter)
On RET/COR/xC: Load instruction offset from O into PC
On RET: Restore PC value from O register (Center right)
On JUMP: Load target offset on DB into PC
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
Input/Output
S Serial IO
Implements SPI functionality with dedicated SER/DES registers
SIR
Serial Input Register
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
(Local Page Index Register)
{LL[7..0]}
{LO[7..0]}
{HI[7..0]}
RAM/ROM
C (Code Page Index)
(Guest-Page Index)
G
Sets BUSY flag, disabling interrupts
Based on working prototype PCB: https://github.com/michaelmangelsdorf/sonne8
Rev. Step 5B
Project Homepage: https://github.com/michaelmangelsdorf/myth
Author: Michael Mangelsdorf
Contact: mimgel-scheider.de
Sheet: /
File: myth.kicad_sch
Title: Sonne-8 Micro-Controller
Size: User Date: 2025-06-14 Rev: Myth
KiCad 8.0.4-9.0.1 Id: 1/1