Myth Microcontroller Reference Schematics

Power

Caps

Clock Phases

~8MHz

"VLSI" Footprint

IO0-7 bus tri-state
internal pull-down

RTS request

Instruction Decoder

Branching

Jump while value register Y not zero
(decrements Y)

Jump always

Jump if R not zero

Jump if R zero

Scrounger

Remap MM, RR, DD etc.
and impracticable opcodes such as NM

Currently NOP,
reserved for
instruction set
extension

Etc.

Preselect decoders

Instruction-Word Latch

Pair DST Decoder

Pair SRC Decoder

SYS decoder

Opcode Type Decoder

8:3 Priority Encoder CD4532

Instruction byte:

all0: OPC_SYS	00000 xxx	See table @ SYS decoder
else: OPC_ADJUST	00001 xxx	b2:extended sign bit b1-0 LSB
OPC_ALU	0001x xxx	See table @ ALU
OPC_TRAP	001xx xxx	b4:MSB b3-0:LSB (remaining bits set 1)
OPC_GETPUT	01xxx xxx	b5-3 OFFS b2:GL b1:GP B0:RO
OPC_PAIR	1xxxx xxx	DEST SRC

GETPUT Decoder

PHA0: Setup PC on address bus
PHA1: Latch opcode into IB0-7, predecoded
PHA2: Decoded, setup source byte on DB or other action
PHA3: Latch source into target or other action
CLOSE: Cleanup

Control Tree

Clear Program Counter
applied during CLOSE

ENTER
Register branch target
CLOSE: Load into C

Inverters

A (A spare register)

B (BROADCAST dummy)

J (JUMP register) (Program Counter)

(Relative addressing)

ADDER

J (the program counter)
is cleared/set to 0
during TRAP or CALL.
Execution starts at offset zero.
The 'head' of the destination page.

Delay CLEAR
until CLOSE

R Adjust

If bits 0-2
not all zero,
sign extend bit 2

Else hard-code
replace 0 by 4

```
0000 => 0100  +4 P4
0001          +1 P1
0010          +2 P2
0011          +3 P3
1100          -4 M4
1101          -3 M3
1110          -2 M2
1111          -1 M1
```

R(esult) Register

All delayed
by PHA2

ALU

0 provides the return offset during RET
receives the return offset during SUBR
can be GET and PUT
provides all address offsets except FETCH
can be stored into
is the implied 2nd ALU input operand

When DC is executed,
prevents the DB output
from being updated with
the new return address
page index, before the
instruction is done.

BLUR3 LUT
ROM layout:

16 maps
@256x256

0 IDR ("identity R" / R:=R)
1 IDO ("identity O" / R:=Old)
2 OCR ("Ones Complement R" / R:=-R)
3 OCO ("Ones Complement O" / R:=-O)
4 SLR ("Shift left New" / R:=R<<1)
5 SLO ("Shift left Old" / R:=O<<1)
6 SRR ("Shift right R" / R:=R>>1)
7 SRO ("Shift right O" / R:=O>>1)

8 AND (R:=R&O)
9 IOR (R:=R|O)
10 EOR (R:=R^O)
11 ADD (Add" / R:=R+O, bits 0-7)
12 CAR ("Carry Bit" / R:= Carry, 9th bit of R+O) 00h or 01h
13 RLO ("Flag: R less than O" / R:= (R<O)? 0xFF:0
14 REQ ("Flag: R equals O" / R:= (R==O)? 0xFF:0
15 RGO ("Flag: R greater than O" / R:= (R>O)? 0xFF:0

O (OFFSET) Operand Register

I (INNER Loop Register)

D (DATA) Page index

C (CODE) Page index

L (LOCAL) Page index (Stack Pointer)

FRO-7 bus
Frame/Record

Automatic call stack
CALL: Enter stack frame
RET: Resume previous frame

Storage

IB 3-5 are the
immediate offset
for GET/PUT
operations

Low

Memory M

HOT switches
between RAM and ROM

Select Program Memory
during Code Phase (IPH)
and Interr Vect

SIR

Serial clock out

SER

SOR

Serial IO

Implements SPI functionality
with dedicated SER/DES registers

Device Enable

Ordinal 0
selects a
NULL DEVICE

Ref. implementation
connects two independent
4-bit groups (high/low)
to 4:16 encoders
(for simultaneous
latch enable
output enable
of two separate devices)

ACTIVE HIGH ENABLE

ACTIVE LOW ENABLE

NULL device

P

PIR

POR

Parallel IO

Implement an 8-bit parallel tri-state
interface bus with dedicated In/Out ports

Glue Logic

Spare Units

Project homepage: https://github.com/Doxflange/Myth
Author: Michael Mangelsdorf
Contact: mikegsk-schaller.de

Sheet: /
File: myth.kicad_sch

Title: Sonne-8 Micro-Controller

Size: A0	Date: 2024-08-17	Rev: Myth Kelvin
KiCad E.D.A. 8.0.4		Id: 1/1