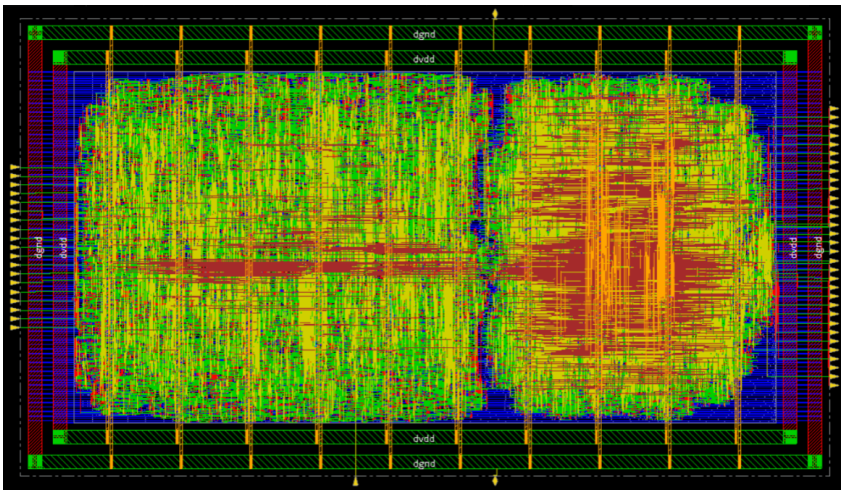
MAS8 v1.0



1 Overview

- Programmable
- 4x 8b Registers
- Custom Assembly
- 16b Instructions
- 100 MHz clock

2 Description

- The *MAS8* is a microprocessor developed entirely in an integrated circuit (IC).
- It is capable of all basic types of instructions: Arithmetic, Logic, Conditionals, Jumps and Memory operations.
- It runs on a custom-made ISA: *MAS*. The instructions are 16 bit wide, with 4 bits reserved for the *opcodes*.
- The operands consist of 4 registers, each with 8 bits.
- The *MAS8* contains 2 independent Memory Blocks, one for data storage (256x32) and another for instruction storage (256x16).
- It supports a programming mode ($\texttt{pr} = 1$), that allows for programming the instruction memory directly.
- The processor is capable of a sustained fetch and execution of 1 instruction per clock cycle. It is not pipelined.
- It also includes an 8 bit counter with load, for the instruction fetch block.
- The ALU is capable of performing arithmetic and logic operations with up to 8 bits, including addition, multiplication and bitwise basic logic operations. More complex operations can be performed as sequences of existing instructions.
- Programs can have up to 256 unique instructions, if the instruction memory is only programmed once. However, if it is programmed in parallel with the code execution, then the aforementioned limit does not apply, as the counter loops back to 0, upon reaching the maximum value.

3 Technical specification

	Unit	Value	Min	Max
Power voltage	V	1.20	1.08	1.32
Temperature	$\circ C$		-40	125
Area	μm^2	343532.796		
Instances		23294		
Worst Negative Setup Slack	ns	0.445		
Worst Negative Hold Slack	ns	-0.000		

4 Pinout

Pin	Type	Dir	Description
pr	Digital	I	Programming mode
en	Digital	I	System enable
rstz	Digital	I	System reset ($\texttt{rstz} = 1 \implies RESET$)
Instr_in <15:0>	Digital	I	Input Instruction (on programming mode only)
reg0 <15:0>	Digital	O	Register R0
reg1 <15:0>	Digital	O	Register R1
reg2 <15:0>	Digital	O	Register R2
reg3 <15:0>	Digital	O	Register R3
clk	Digital	I	Clock signal
dvdd	Power	I/O	Digital supply
dgnd	Power	I/O	Digital ground

5 Detailed Description

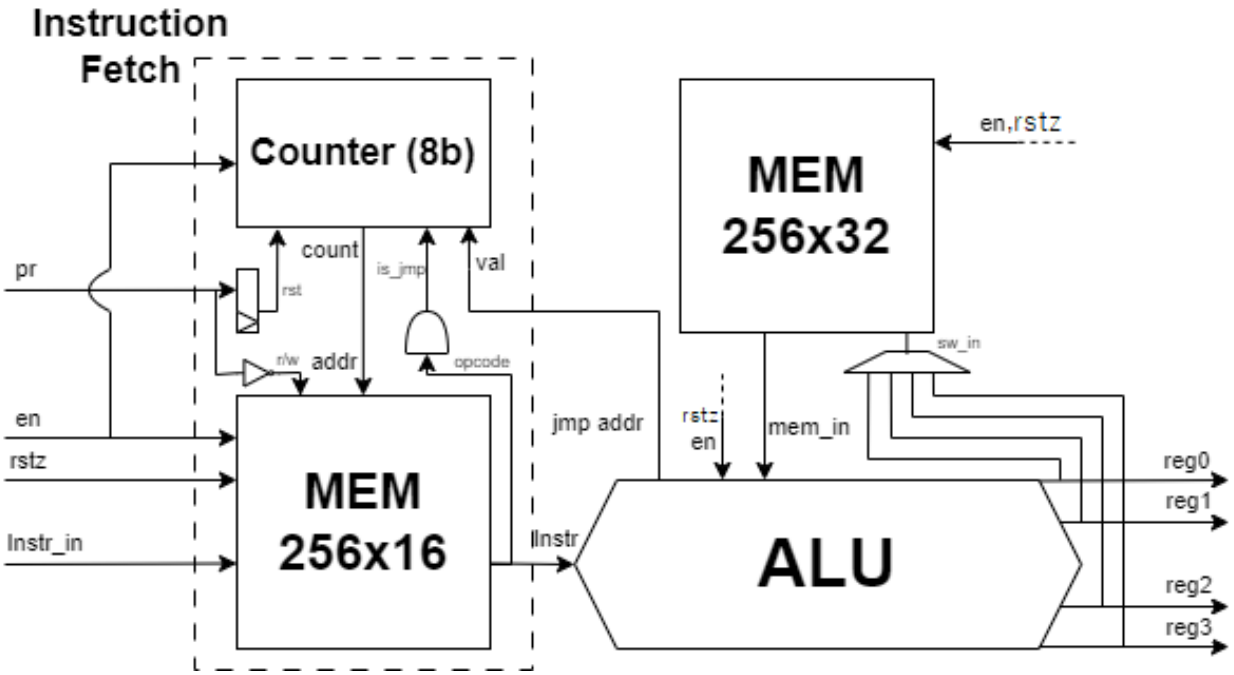
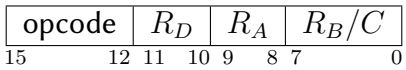


Figure 1: *MAS8* block diagram.

5.1 MAS

The *MAS8* runs on a custom-made instruction set architecture: *MAS*.
This ISA contains 13 unique instructions, each 16 bit wide. All instruction codes are built in the following order:



- opcode: Operation Code (4 bits)
- R_D: Destination Register ID (2 bits)
- R_A: Operand A Register ID (2 bits)
- R_B/C: Operand B Register ID (2 bits, but represented with 8 bits) or 8 bit immediate (constant, C)

Operand Registers: R0 (00), R1 (01), R2 (10), R3 (11)
Others: PC (Program Count), C (Immediate Constant)

Opcode	Instruction	Arguments	RTL Description	Description
0000	ADC	R_D, R_A, C	$R_D \leftarrow R_A + C$	Add Constant
0001	ADD	R_D, R_A, R_B	$R_D \leftarrow R_A + R_B$	Add
0010	MUL	R_D, R_A, R_B	$R_D \leftarrow R_A * R_B$	Multiply
0011	SRA	R_D, R_A, R_B	$R_D \leftarrow R_A >> R_B$	Shift Right Arithmetic
0100	AND	R_D, R_A, R_B	$R_D \leftarrow R_A \,\&\, R_B$	Bit-wise And
0101	OR	R_D, R_A, R_B	$R_D \leftarrow R_A \mid R_B$	Bit-wise Or
0110	NOT	R_D, R_A	$R_D \leftarrow !R_A$	Bit-wise Not
0111	XOR	R_D, R_A, R_B	$R_D \leftarrow R_A \oplus R_B$	Bit-wise Xor
0111	LW	R_D, R_A, C	$R_D \leftarrow MEM[R_A + C]$	Load Word
0111	SW	R_D, R_A, C	$MEM[R_A + C] \leftarrow R_D$	Store Word
0111	LT	R_D, R_A, R_B	$R_D \leftarrow R_A < R_B \,?\, 1 : 0$	Less Than
0111	LTC	R_D, R_A, C	$R_D \leftarrow R_A < C \,?\, 1 : 0$	Less Than Constant
0111	J	R_A, C	$PC \leftarrow PC + (R_A + C)$	Jump

Examples:

- ADD R0 , R1 , R2 = 1102h $\implies$ R0 = R1 + R2
- ADC R1 , R1 , 4 = 0504h $\implies$ R1 = R1 + 4
- NOT R3 , R2 = 6E00h $\implies$ R3 = !R2
- SW R2 , R3 , 0 = 9B00h $\implies$ Store R2 in memory, on address R3 + 0
- J R0 , 0 = F000h $\implies$ PC = PC + R0 + 0 (jump R0 + 0 instructions ahead)

5.2 Instruction Fetch (IF)

The Instruction Fetch (IF) block responsible for fetching the next instruction to execute, as the name implies. It is composed of 2 main blocks:

- Counter (8 bits)
- Memory Block (256 x 16b)

All instructions of a certain program are stored in the memory block, 1 instruction per line.
Specifications:

- It is possible to program the instructions directly to memory, by turning the `pr` digital signal on and injecting each instruction, sequentially, into the `instr_in` digital bus.
- Once the programming is done, it is suggested to force all bus bits to 0, since the instruction code 0000h represents a NOP (No-operation).
- To run the code afterwards, simply turn `pr` off. To reset the memory, turn `rstz` **off**. To pause the program, disable the system by turning `en` off.

The IF block outputs the 16 bit instruction that is to be executed, which connects directly to the ALU block.

5.3 Arithmetic Logic Unit (ALU)

It is able to perform 8 bit addition, multiplication, arithmetic shift right and bit-wise logic operations (and, or, not and xor).
Specifications:

- The ALU executes only 1 instruction at a time. There are no pipelining stages in any of the processor's blocks.
- It outputs the values of all 4 operand registers (R0-3), as well as the jump address, for the IF block, and the memory address, for loading or storing register values.
- The `rstz` digital signal also resets the values of all 4 operands, with the default (reset) value being 00h. Likewise, the `en` signal also disables the ALU, so that the user is able to pause the program execution.

5.4 Data Memory (DMEM)

All store operations store the specified values in the Data Memory (DMEM).
It contains 256 lines, each with 32 bits. However, as the address only specifies the line and no logic to decode the column was implemented, all words are stored in the last 8 bits of each line. (tbc)
Specifications:

- As with the previous blocks, the DMEM can be enabled and disabled, with `en` and reset, with `rstz`.
- It receives the input value to be stored, along with the specified address.
- It is also possible to read from DMEM, to perform a load operation. As such, the DMEM outputs the value to be loaded, according to the specified address.

6 Assembly Guidelines

Pin	Recomendation
pr	Digital signal. No special recommendation
en	Digital signal. No special recommendation
rstz	Digital signal. No special recommendation
Instr_in <15:0>	Digital signal. No special recommendation
reg0 <15:0>	Digital signal. No special recommendation
reg1 <15:0>	Digital signal. No special recommendation
reg2 <15:0>	Digital signal. No special recommendation
reg3 <15:0>	Digital signal. No special recommendation
clk	Consider as a very noisy signal
dvdd	Consider as noisy supply signal
dgnd	Consider as noisy supply signal

7 Full Chip Layout

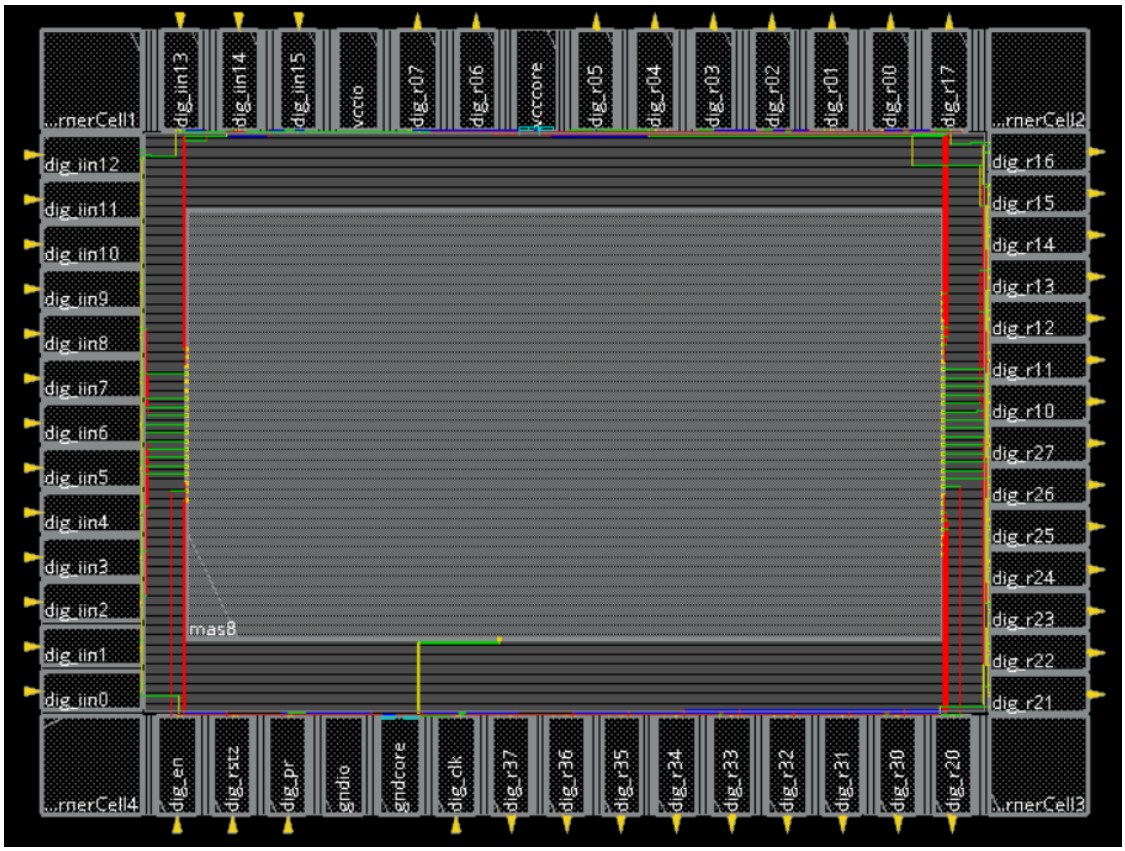


Figure 2: Full Chip Layout.

8 Version History

Version	Date	Description
MAS8 v1.0	March 2024	First version of the MAS8.