

# CatCore-Processor 2.0

Bruno Ogata

July 18, 2018

[github.com/brunoogata/CatCORE-Processor](https://github.com/brunoogata/CatCORE-Processor)

## 1 Proposta

O processador CatCore foi baseado na arquitetura MIPS monociclo em que cada instrução só pode ser executada em um ciclo de clock. Sua unidade de processamento contém como mencionado um Banco de Registradores, para armazenar os dados a processar, uma Unidade Lógica Aritmética, para realizar as operações lógico-aritméticas, dois somadores e quatro multiplexadores, para controle, além de um Program Counter, para definir o endereço da instrução a ser executada. A memória principal é composta de uma Memória de Instruções, que armazena as instruções a serem executadas, e uma Memória Principal, para armazenar informações que não estão presentes no Banco de Registradores. Para controlar a ordem que as operações nos módulos são executadas, foi implementada uma Unidade de Controle e como interface para interação com o usuário, foi desenvolvido um módulo de Entrada-Saída.

## 2 Conjunto de Instruções

Conjunto de Instruções formado por 25 instruções envolvendo operações aritméticas, deslocamento de memória e pulos de endereços.

## 2.1 Instruções e OPCODEs

Figure 1: Instruções com seus respectivos opcodes

<b>ADD</b>	100000	<b>ADDI</b>	010000	<b>J</b>	111111
<b>SUB</b>	100001	<b>SUBI</b>	010001	<b>JR</b>	011010
<b>MULT</b>	101010	<b>LOADI</b>	011001	<b>JAL</b>	011000
<b>DIV</b>	101011	<b>LW</b>	011111	<b>HALT</b>	111110
<b>SMLEQ</b>	100100	<b>SW</b>	011110	<b>NOP</b>	000000
<b>SML</b>	100101	<b>LWR</b>	011101	<b>FUNCTION</b>	000000
<b>LGREQ</b>	100110	<b>SWR</b>	011100	<b>LABEL</b>	000000
<b>LGR</b>	100111	<b>BNE</b>	010111	<b>INPUT</b>	000111
<b>EQ</b>	101110	<b>MOV</b>	010100	<b>OUTPUT</b>	111000

## 2.2 Mapeamento das Instruções

Instrução	OpCode	Modo de Endereçamento	Operação	Tipo
ADD	100000	Por Registrador	$RD = RS + RT$	R
ADDI	010000	Imediato	$RT = RS + IMED$	I
SUB	100001	Por Registrador	$RD = RS - RT$	R
SUBI	010001	Imediato	$RT = RS - IMED$	I
MULT	100000	Por Registrador	$RD = RS * RT$	R
MULTI	100000	Imediato	$RD = RS * IMED$	I
DIV	100000	Por Registrador	$RD = RS / RT$	R
DIVI	100000	Imediato	$RD = RS / IMED$	I
AND	100010	Por Registrador	$RD = RS \& RT$	R
OR	100011	Por Registrador	$RD = RS \mid RT$	R
SML	100100	Por Registrador	$RD = (RS \ll RT)$	R
SMLEQ	100011	Por Registrador	$RD = (RS \ll RT)$	R
LGR	100011	Por Registrador	$RD = (RS \ll RT)$	R
LGREQ	100011	Por Registrador	$RD = (RS \ll RT)$	R
EQ	100011	Por Registrador	$RD = (RS == RT)$	R
MOV	010100	Por Registrador	$RT = RS$	R
BEQ	010110	Relativo ao PC	condicional ==	I
BNQ	010111	Relativo ao PC	condicional !=	I
SW	011110	Direto	$Mem[AD] = RT$	I/O
LW	011111	Direto	$RT = Mem[AD]$	I/O
LOADI	011001	Imediato	$RT = IMED$	I/O
LWR	011111	Direto	$RT = Mem[RD]$	I/O
SWR	011111	Direto	$Mem[RD] = RT$	I/O
J	111111	Absoluto	Jump	J
NOP	000000		$PC + 1$	J
HALT	111110		Espera ENTER	J
IN	000111		$RT = SWITCHES(E/S)$	I/O
OUT	111000		$DISPLAY(E/S) = RT$	I/O

### 3 Arquitetura Base

Figure 2: Arquitetura base do CatCore

