ARCHITECTURE SPECIFICATION

32-bit uDLX Core Processor

Universidade Federal da Bahia

Versão: 1.0

GNU LGPL License

Histórico de Revisões

Date	Description	Author(s)
04/27/2014	Conception	João Carlos Bittencourt
04/30/2014	Instruction layout description	João Carlos Bittencourt

CONTENTS

1. Introduction

1.1. Purpose

The main purpose of this document is to define specifications of a uDLX implementation and to provide a full overview of the design. This specifications defines all implementation parameters that composes the general uDLX requirements and specification. This definitions include processor operation modes, instruction set (ISA) and internal registers characteristics. This document also include detailed information of pipeline stages architecture, buses and other supplemental units.

1.2. Document Outline Description

This document is outlined as follow:

- Section :
- Section :

1.3. Acronyms and Abbreviations

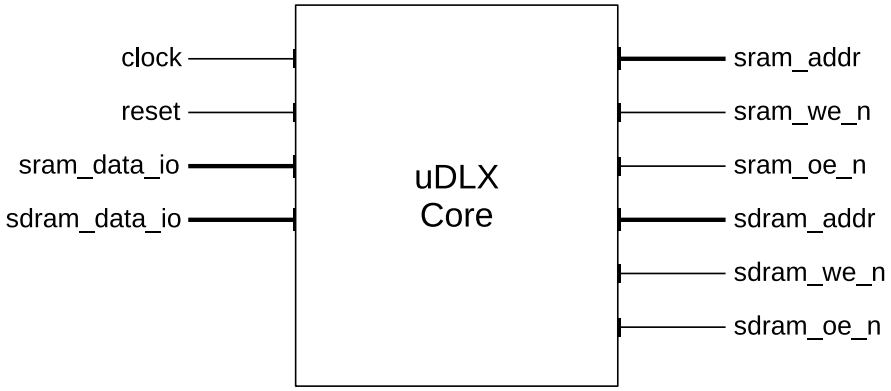
Along this and other documents part of this project, it will be recurrent the usage of some acronyms and abbreviations. In order to keep track of this elements the Table 1 presents a set of abbreviations used and its corresponding meaning.

Table 1: Acronym and descriptions of elements in this document.

Acronym	Description
RISC	Reduced Instruction Set Computer
GPR	General Purpose Registers
FPGA	Field Gate Programmable Array
GPPU	General Purpose Processing Unit
SDRAM	Synchronous Dynamic Random Access Memory
HDL	Hardware Description Language
RAW	Read After Write
CPU	Central Processing Unit
ISA	Instruction Set Architecture
ALU	Arithmetic and Logic Unit
PC	Program Counter
RFlags	Flags Register
Const	Constant

2. Architecture Overview

2.1. Block Diagram



2.2. Pin/Port Definitions

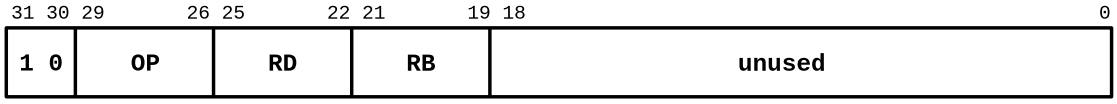
Name	Length	Direction	Description
clock	1	input	CPU core clock
reset	1	input	CPU core reset
sram_data_io	16	in/out	SRAM data
sdram_data_io	32	in/out	SDRAM data
sram_addr	20	input	SRAM address
sram_we_n	1	output	SRAM write enable
sram_oe_n	1	output	SRAM output enable
sdram_addr	13	in/out	SDRAM address
sdram_we	1	output	SDRAM write enable
sdram_oe	1	output	SDRAM output enable

2.3. Parameters and Configurations

Name	Value	Description

3. Instructions Layout

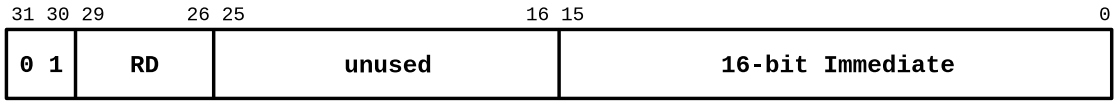
3.1. ALU



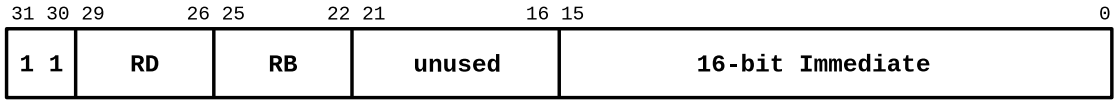
OP	Opperation	Mnemonic	Flags Update
0000	$R_D = R_D + R_F$	add d, f	all
0001	$R_D = R_D - R_F$	sub d, f	all
0010	$R_D = R_D * R_F$	mul d, f	all
0011	$R_D = R_D / R_F$	div d, f	all
0100	$R_D = R_D \; and \; R_F$	and d, f	above, equal, error
0101	$R_D = R_D \; or \; R_F$	or d, f	above, equal, error
0110	$R_flags = R_D \; cmp \; R_F$	cmp d, f	above, equal, error
0111	$R_D = \; not \; R_D$	not d	above, equal, error

3.2. Immediate

Type I



Type II



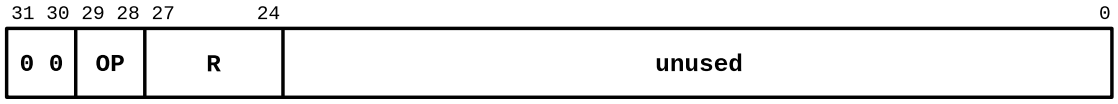
Type	Opperation	Mnemonic
I	$R_D = I_{16}$	load immediate, d
II	$R_D = [I_{16} + R_B]$	load immediate, d, b
II	$[I_{16} + R_B] = R_D$	load d, immediate, b

3.3. Control Transfer

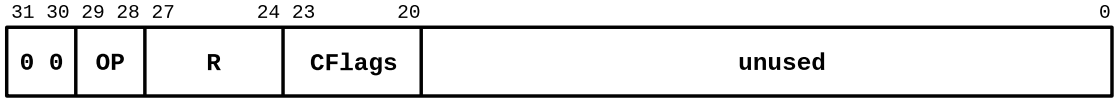
The μDLX core processor has five control transfer instructions encoded using the following three types. The first encoding type is used for unconditional jump and subroutine

call. The second one is used for conditional branch, based on ALU flags. The third one reffers to the unconditional jump related to PC by an immediate value offset.

Type I



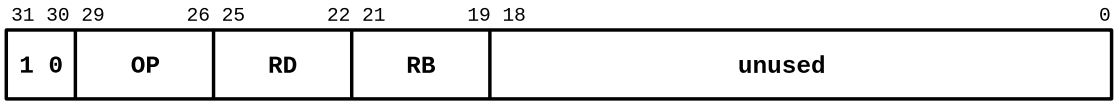
Type II



Type III

Type	OP	Opperation	Mnemonic
I	00	Jump Register	jr r
I	01	Subroutine call	call r
II	10	Branch flags	brfl r, const
III	11	Jump PC	jpc destiny

3.4. Memory

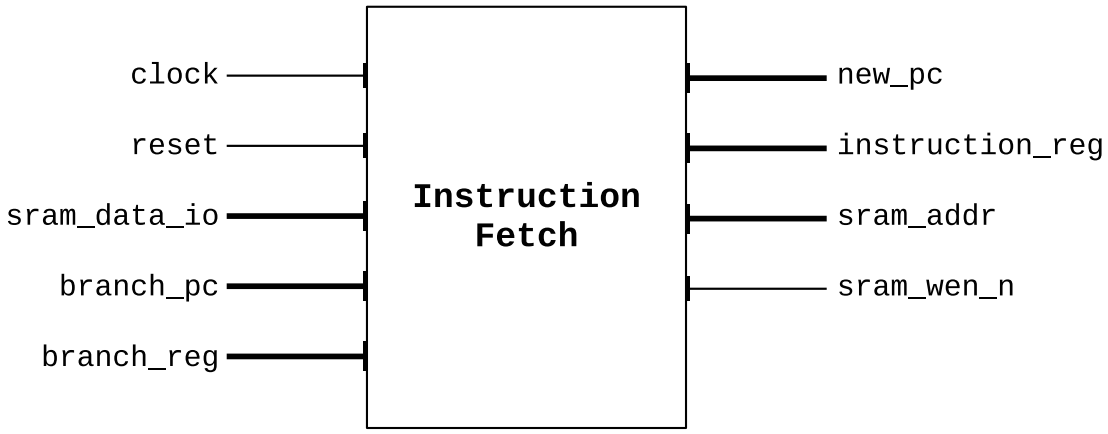


OP	Opperation	Mnemonic
1000	$R_D = Mem[R_B]$	load d, b
1100	$Mem[R_B] = R_D$	store b, d

4. Architecture Description

4.1. Instruction Fetch

4.1.1. *Block Diagram*

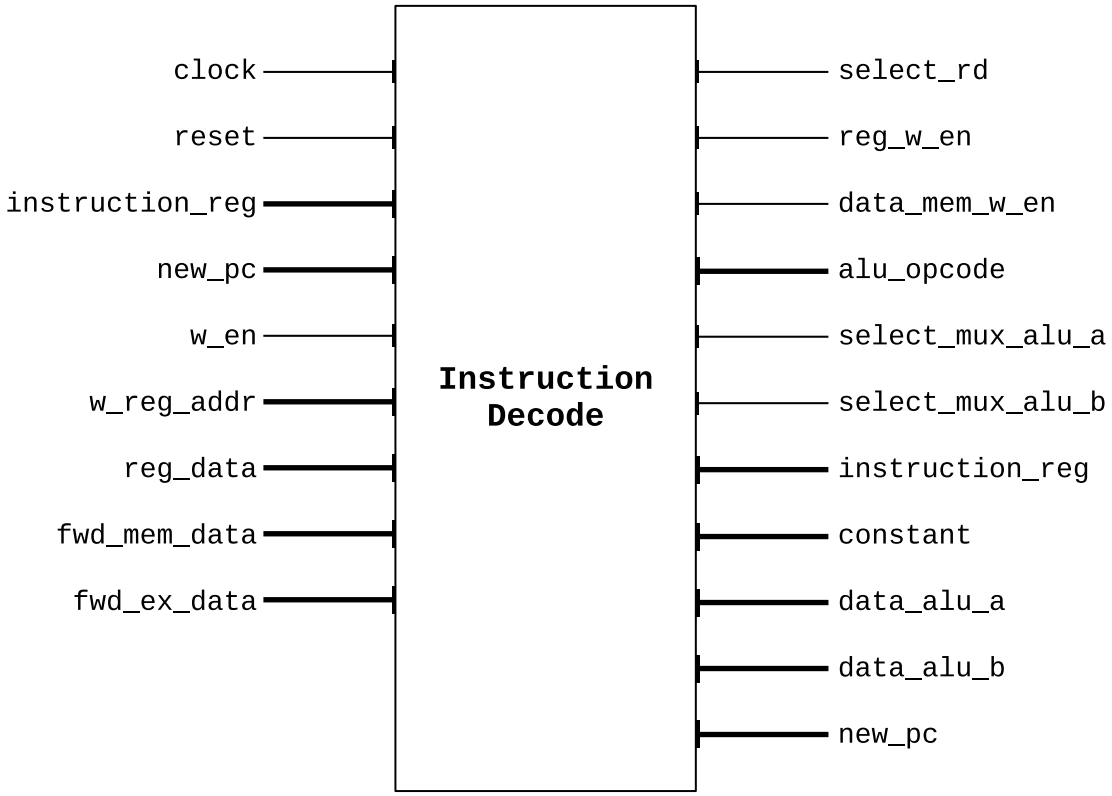


4.1.2. *Pin/Port Definitions*

Name	Length	Direction	Description
clock	1	input	CPU core clock
reset	1	input	CPU core reset
sram_data_io	16	in/out	SRAM data
branch_pc	20	input	Branch address PC relative
branch_reg	20	input	Branch address loaded from registers
npc	20	output	New program counter value
instruction	32	output	CPU core instruction
sram_addr	20	output	SRAM address
sram_we	1	output	SRAM write enable

4.2. Instruction Decoding

4.2.1. Block Diagram



4.2.2. Pin/Port Definitions

Name	Length	Direction	Description
clock	1	input	CPU core clock
reset	1	input	CPU core reset
instruction_reg	32	input	SRAM data
new_pc	20	input	Branch address PC relative
w_en	1	input	Branch address loaded from registers
w_reg_addr	4	input	New program counter value
reg_data	32	input	CPU core instruction
fwd_mem_data	32	input	SRAM address
fwd_ex_data	32	input	SRAM write enable
select_rd	TBD	output	SRAM output enable
reg_w_en	TBD	output	

continued on next page

continued from previous page			
Name	**Length**	**Direction**	**Description**
data_mem_w_en	1	output	
alu_opcode	3	output	
select_mux_alu_a	TBD	output	
select_mux_alu_b	TBD	output	
instruction_reg	32	output	
constant	32	output	
data_alu_a	32	output	
data_alu_b	32	output	
new_pc	20	output	

4.3. Stage 3 - Execution

4.4. Stage 4 - Memory

4.5. Stage 5 - Write Back

4.6. Pipeline Register Description

4.7. Control Micro-instructions Description