ipPROCESS

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 **5**

1.1 Purpose 5

1.2 Document Outline Description 5

1.3 Acronyms and Abbreviations 5

2 Architecture Overview **6**

2.1 Block Diagram 6

2.2 Pin/Port Definitions 6

2.3 Parameters and Configurations 6

3 Instructions Layout **7**

3.1 ALU 7

3.2 Immediate 7

3.3 Control Transfer 7

3.4 Memory 8

4 Architecture Description **9**

4.1 Instruction Fetch 9

4.1.1 Block Diagram 9

4.1.2 Pin/Port Definitions 9

4.2 Instruction Decode/Register Fetch 10

4.2.1 Block Diagram 10

4.2.2 Pin/Port Definitions 10

4.3 Execute/Address Calculate 11

4.4 Memory Access 11

4.4.1 Block Diagram 11

4.4.2 Pin/Port Definitions 11
4.5 Write Back 12
4.5.1 Block Diagram 12
4.5.2 Pin/Port Definitions 12
4.6 Pipeline Register Description 13
4.7 Forwarding Unit 13
4.8 Branch Prediction Buffer 13
4.9 Control Micro-instructions Description 13

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

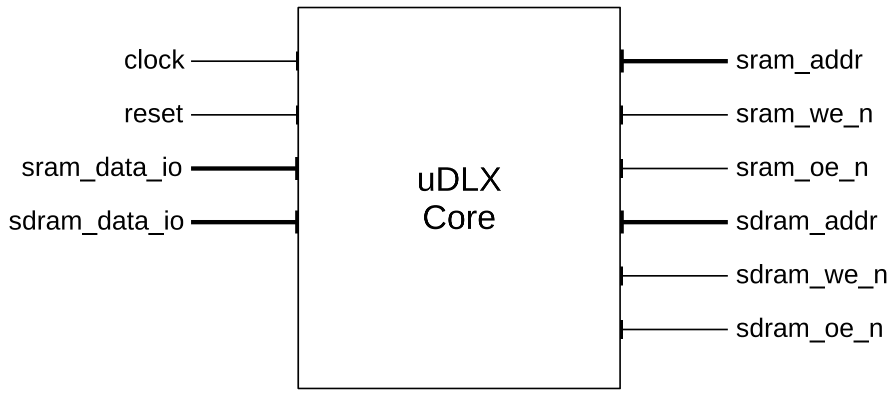
Along this and other documents part of this project, it will be recurrent the usage of some acronyms and abbreviations. In order to keep track of this elements the Table 1 presents a set of abbreviations used and its corresponding meaning.

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

31 30	29 26	25 22	21 19	18 0
1 0	OP	RD	RB	unused

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

31 30	29 26	25 16	15 0
0 1	RD	unused	16-bit Immediate

Type II

31 30	29 26	25 22	21 16	15 0
1 1	RD	RB	unused	16-bit Immediate

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

31 30	29 28	27 – 24	0
0 0	OP	R	unused

Type II

31 30	29 28	27 – 24	23 – 20	0
0 0	OP	R	CFlags	unused

Type III

31 30	29 28	27 – 0
0 0	OP	PC-relative destiny

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

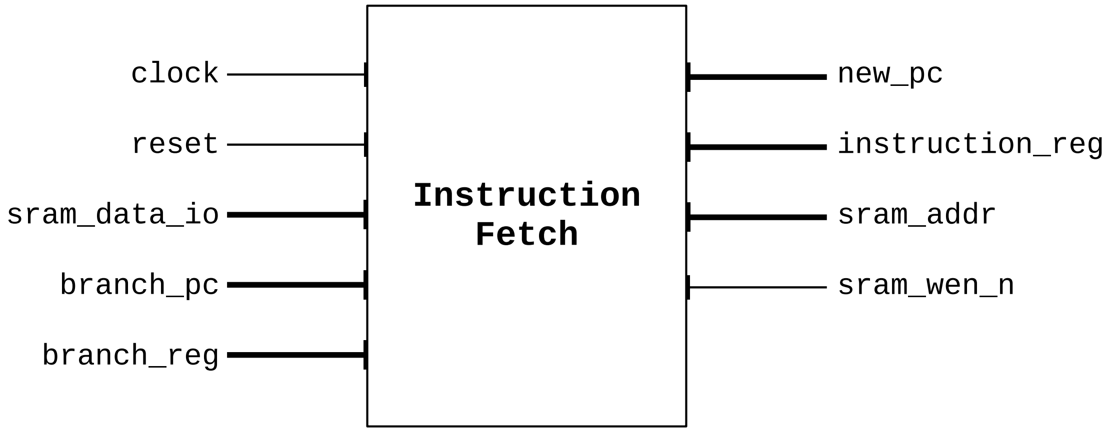
31 30	29 – 26	25 – 22	21 – 19	18 – 0
1 0	OP	RD	RB	unused

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

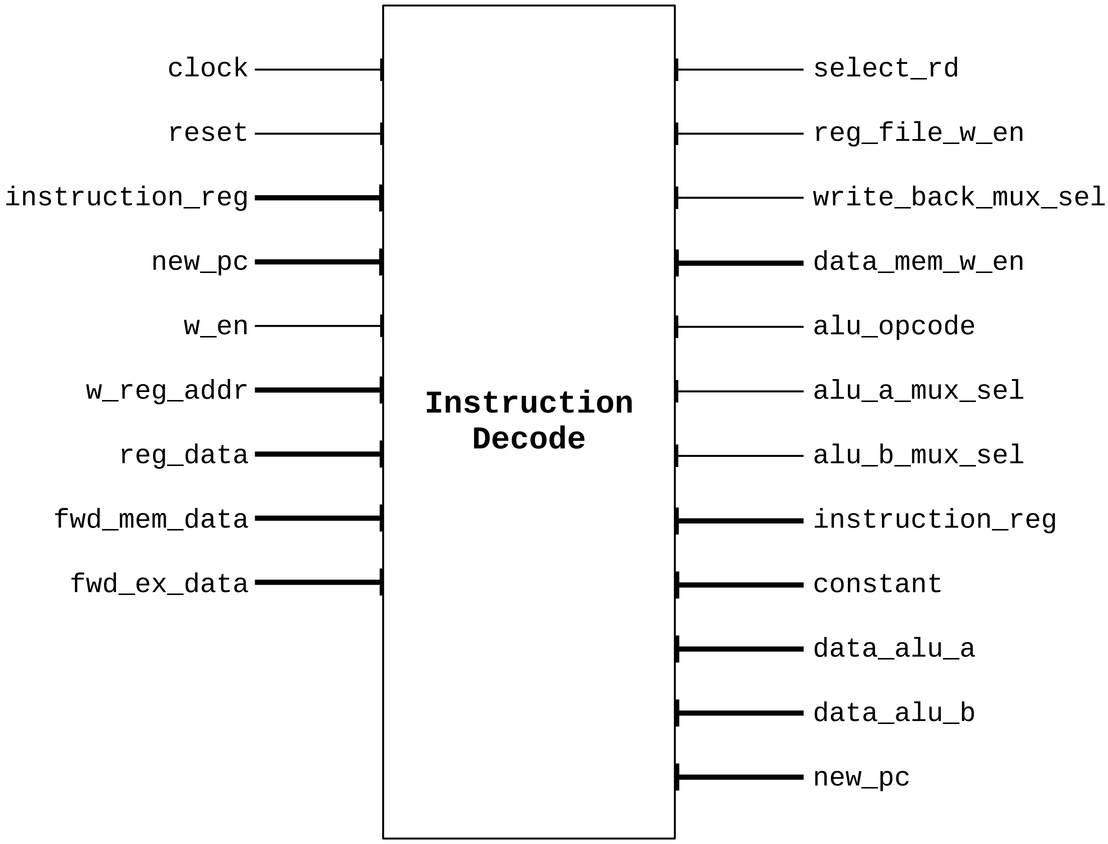


4.1.2. Pin/Port Definitions

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
new_pc	20	output	Updated value of PC
instruction	32	output	CPU core instruction
sram_addr	20	output	SRAM address
sram_we	1	output	SRAM write enable

4.2. Instruction Decode/Register Fetch

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	CPU core instruction
new_pc	20	input	Updated value of PC
w_en	1	input	GPR bank write enable signal
w_reg_addr	4	input	GPR bank destiny address
reg_data	32	input	GPR bank write data
fwd_mem_data	32	input	Forwarding data from DRAM output
fwd_ex_data	32	input	Forwarding data from ALU output
select_rd	TBD	output	TBD

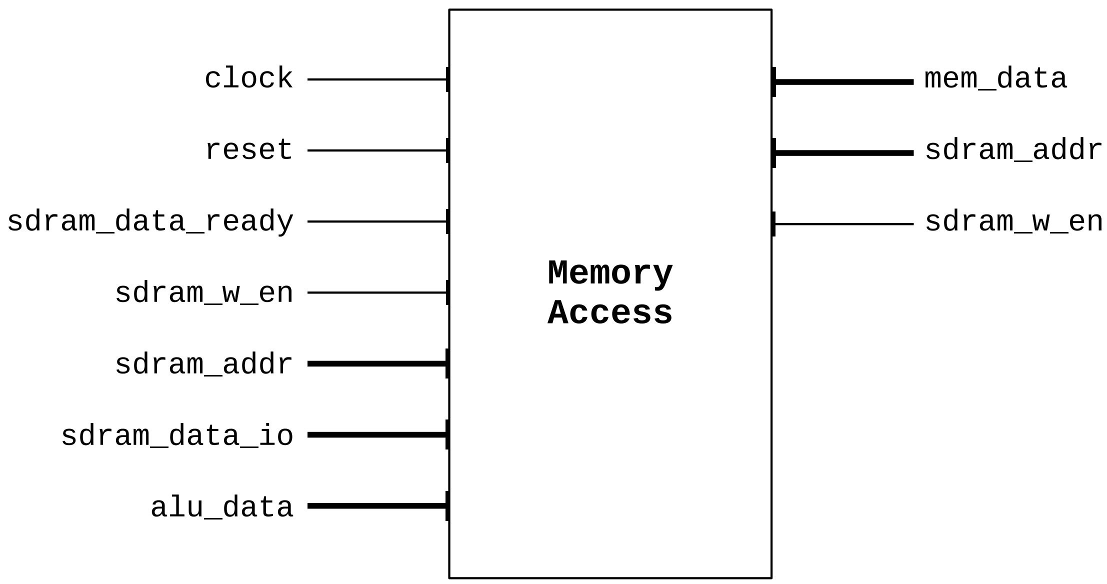
continued on next page

continued from previous page			
Name	**Length**	**Direction**	**Description**
reg_file_w_en	TBD	output	GPR bank write enable
write_back_mux_sel	TBD	output	Write back mux select
data_mem_w_en	1	output	SDRAM write enable
alu_opcode	3	output	ALU opperation code
select_mux_alu_a	TBD	output	ALU input A data select
select_mux_alu_b	TBD	output	ALU input B data select
instruction_reg	32	output	CPU core instruction
constant	32	output	32-bit Sign-extended constant
data_alu_a	32	output	ALU input A data
data_alu_b	32	output	ALU input B data
new_pc	20	output	Updated value of PC

4.3. Execute/Address Calculate

4.4. Memory Access

4.4.1. Block Diagram

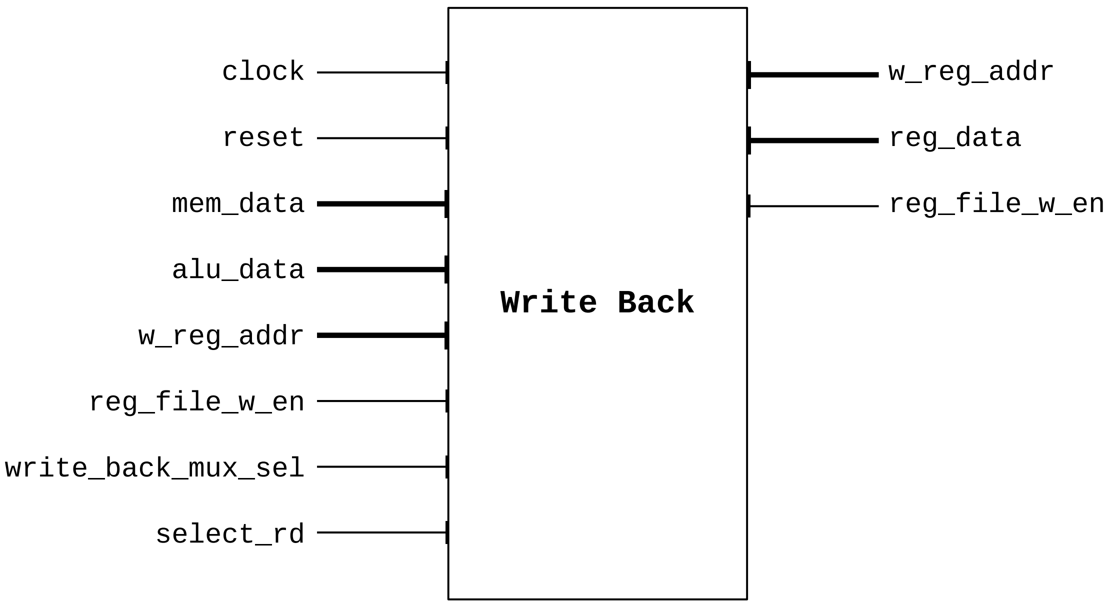


4.4.2. Pin/Port Definitions

Name	Length	Direction	Description
clock	1	input	CPU core clock
reset	1	input	CPU core reset
sdram_dara_ready	1	input	SDRAM data ready control
sdram_w_en	1	input	SDRAM write enable
sdram_addr	13	input	SDRAM read/write address
sdram_data$_{io}$	32	input	SDRAM I/O data
alu_data	32	input	ALU data output
mem_data	32	output	Memory output data
sdram_addr	13	output	SDRAM read/write address
sdram_w_en	1	output	SDRAM write enable

4.5. Write Back

4.5.1. *Block Diagram*



4.5.2. *Pin/Port Definitions*

Name	Length	Direction	Description
clock	1	input	CPU core clock
reset	1	input	CPU core reset

continued on next page

continued from previous page			
Name	**Length**	**Direction**	**Description**
mem_data	32	input	SDRAM data output
alu_data	32	input	ALU data output
w_file_w_en	4	input	GPR bank write enable signal
w_reg_addr	1	input	GPR bank destiny address
write_back_mux_sel	TBD	input	Write back mux select
select_reg_data	TBD	input	Select data to be writen in GPR bank
w_reg_addr	4	output	GPR bank destiny address
reg_data	32	output	GPR bank write data
reg_file_w_en	1	output	GPR bank write enable signal

4.6. Pipeline Register Description

4.7. Forwarding Unit

TBD in further releases.

4.8. Branch Prediction Buffer

TBD in further releases.

4.9. Control Micro-instructions Description