Branch Prediction Game

<Team Name>

Brody Skaleski, …, …, …

**University of Wisconsin-Madison**

Project Proposal Document

Table of Contents

[1. Introduction 2](#_Toc127865136)

[2. Hardware Block Diagram 3](#_Toc127865137)

[2.1. << Block X Description >> 3](#_Toc127865138)

[3. Processor 3](#_Toc127865139)

[3.1. ISA Summary 3](#_Toc127865140)

[3.2. Condition Codes 4](#_Toc127865141)

[3.3. Addressing Modes 4](#_Toc127865142)

[4. Software Blocks 4](#_Toc127865143)

[4.1. Assembler/Compiler 5](#_Toc127865144)

[4.2. Simulator 5](#_Toc127865145)

[4.3. Application 5](#_Toc127865146)

[5. Division of Labor 5](#_Toc127865147)

# Introduction

This document consolidates the project that your team has decided on. Following guidelines might help in achieving a smooth execution

1. You as a team have done enough brainstorming on project ideas
2. Everyone in the team has a good idea about their role in the project and are actively participating in the project
3. Software – assembly, GUI, programs, simulators are a part of the project. Include them in planning at all the stages and documentation is required for all of them
4. Does your team have a file sharing methodology worked out (GitHub or other). Keep repos private.

Ideally, this document has to contain the following information about your project. The Introduction section will provide a brief overview of your project along with the inspiration and ideas behind this. The initial submitted document is a representative of your commitment to the project and will be used to evaluate the project. Update the contents page before submitting (right click and update).

Before submitting this document, all members in the team have to sign it to acknowledge that they have read this document and understand the project components and expectations.

# Hardware Block Diagram

<<Present the hardware block diagram here>>

<< Block Diagram should be followed by a high level overview of function>>

If your project requires you to show that software is playing a key role in controlling the hardware then describe its function.

## << Block X Description >>

For each block in the block diagram give a more detailed description of its function. If it will contain memory mapped control registers then provide a table describing the control register function, and assigning an address to it. You will not be held to these specs, but want to see that you gave them some thought.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mneumonic | Encoding | Sample Instruction | Encoding | Explanation | Comments |
| Add |  | Add R3, R2, R1 |  | R3 <= R2 + R1 |  |
| Addz |  | ADDZ R6, R5, R4 |  |  |  |
| Sub |  | Sub R9, R8, R7 |  | R9 <= R8 – R7 |  |
| And |  | And R12, R11, R10 |  | R12 <= R11 & R10 |  |
| nor |  | Nor R10, R12, R13 |  | R10 <= ~(R12|R13) |  |
| Sll |  | Sll R1, R0, 14 |  | R1 <= R0 << 14 |  |
| Sra |  | Srl R3, R2, 1 |  | R3 <= R2 >> 1 |  |
| Lw |  | Lw R7, R6, 5 |  | R7 <= mem[R6+5] |  |
| Sw |  | SW R12, R14, 13 |  | Mem[R14 +13] <= R12 |  |
| Lhb |  | Lhb R13, 12 |  | R13 <= |  |
| Llb |  | Llb R12, 11 |  | R12 <= sign-extend{11} |  |
| ANDN |  | Andn R12, R13, 1 |  |  |  |
|  |  |  |  |  |  |
| Branch |  |  |  |  |  |
| Neq |  | b neq, label |  | Branch if Z=0 |  |
| Eq |  | B eq, label |  | Branch if Z=1 |  |
| Gt |  | B gt, label |  | Branch if {Z,N}==2'b00 |  |
| Lt |  | B lt, label |  | Branch if N=1 |  |
| Gte |  | B gte, label |  | Branch if N=0 |  |
| Lte |  | B lte, label |  | Branch if N=1 or Z=1 |  |
| Ovfl |  | B ovfl, label |  | Branch if V=1 |  |
| Uncond |  | B uncond, label |  | Branch unconditionally |  |
|  |  |  |  |  |  |
| Jump |  |  |  |  |  |
| Jal |  | Jal label |  |  |  |
| Jr |  | Jr R15 |  |  |  |
| Hlt |  | Hlt |  |  |  |
|  |  |  |  |  |  |
| Immediate |  |  |  |  |  |
| ADDI |  | Add R3, R2, IMM |  |  |  |
| SUBI |  | Sub R13, R12, IMM |  |  |  |
| XORI |  | Xor R4, R5, IMM |  |  |  |
| ANDNI |  | Andni R7, R8, IMM |  |  |  |
| ANDI |  | ANDI, R3, R8, IMM |  |  |  |
|  |  |  |  |  |  |
| MOVC |  | Movc R12, R3 |  |  |  |
| MUL |  | Mul R8, R5, R4 |  |  |  |
| Push |  | Push R14 |  |  |  |
| pop |  | POP R14 |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## Processor

Your processor is a critical component of your project. As with other components of the project, you need to design your CPU appropriately. You **must** fill in all the subsections in the CPU section.

### ISA Summary…I can do this part if we use my 552 ISA(brody)

| **mneumonic** | **Encoding** | **Sample Instruction** | **Encoding** | **Explanation** | **Other Comments** |
| --- | --- | --- | --- | --- | --- |
| ADD | 00000 dddd ssss tttt 000 | ADD R2, R4, R7 | 0x01238 | R2 🡨 R4 + R7 | All flags updated |
| ADDC | 00001 dddd ssss tttt 000 | ADDC R1, R9, R2 | 0x08C90 | R1 🡨 R9 + R2 + C | All flags updated |
| SUB | 00010 dddd ssss tttt 000 | SUB R8, R3, R4 | 0x141A0 | R8 🡨 R3 + R4 | All flags updated |
| SUBB | 00011 dddd ssss tttt 000 | SUBB R11, R4, R3 | 0x1DA18 | R11 🡨 R4 – R3 - C̅ | All flags updated |
| AND | 00100 dddd ssss tttt 000 | AND R2, R4, R7 | 0x21238 | R2 🡨 R4 & R7 | N & Z flags updated |
| OR | 00101 dddd ssss tttt 000 | OR R1, R9, R2 | 0x28C90 | R1 🡨 R9 | R2 | N & Z flags updated |
| NAND | 00110 dddd ssss tttt 000 | NAND R8, R3, R4 | 0x341A0 | R8 🡨 ~(R3 & R4) | N & Z flags updated |
| XOR | 00111 dddd ssss tttt 000 | XOR R11, R4, R3 | 0x3EA18 | R11 🡨 R4 ^ R3 | N & Z flags updated |
| ADDI | 01000 dddd ssss llll lll | ADDI R12, R9, 0x01 | 0x46481 | R12 🡨 R9 + 1 | All flags updated |
| ADDIEQ | 01001 dddd ssss llll lll | ADDIEQ R3, R1, 3 | 0x49883 | R12 🡨 R1 + 3 only if Z = 1 | Flags conditionally updated |

A table similar to show below should be filled out for your ISA. This section should also include other info about your ISA. Like datapath width, instruction width, and any registers in the RF that have special functions (like link register or SP).

### Condition Codes

### Addressing Modes

List and explain all addressing modes supported by your ISA.

#### Immediate (will use)

If immediate operands are supported by your ISA what are the different forms and how are they handled

#### Harvard vs Von Neumann

We will be using the Hazard Architecture. Accessing data from the instruction memory will be done by…………………………

If Harvard describe your support for accessing data from instruction memory

#### Special Features

One special performance feature of our ISA is predict not-taken dynamic branch predication. We are implementing this to improve the instruction flow through the processor’s pipeline.

Chart

Description automatically generatedOur second special performance feature is Floating point support. Any mathematical operation that can be completed using our ALU (described in our ISA) will support floating point values. We are implementing this using the IEEE 754 floating-point format of 1-bit sign indicator, a 5-bit biased exponent, and a 10-bit fraction, such as the image below.

#### Co-processors

Does your processor or project support a co-processor or dedicated functional block (floating point, encryption engine, FFT, …). If so describe here the function of the block and how the processor will interface with it.

#### Other architectural features

If there are other architectural features of significance (SIMD, VLIW, special purpose processor) describe them here

# Software Blocks

Mention all the software blocks present in your project. These can be one or more of the following.

1. Assembler
2. Compiler
3. Simulator
4. Application

You are required to describe each software block in sufficient detail.

## Assembler/Compiler

Assembler has to set expectations for the assembly code.

## Simulator

If a simulator is used in your project, explain the inputs and outputs for your simulator and how that will help in your project.

## Application

Your application should show the strengths of the hardware design. Explain how your application software will demonstrate your project and the hardware features. If you have a GUI in mind, you can put it down in this document. Application interface with the external world and the hardware has to be defined. If the application is expecting some memory space to have some functionality, the expectations have to be documented.

# Division of Labor

Provide a table indicating high level tasks that different team members will be responsible for. You will not be held to this initial division of labor, but we need to know you have at least give it some thought.