ELEC6025: VLSI Design Project
Part 1: Microprocessor Research
Topic: Test and Branch

Henry Lovett
Team: R4
Course Tutor: Mr B. Iain McNally

February 6, 2014

Contents

1 Introduction

This report details the research done for the “Test and Branch” part of team R4 for the VLSI design module (ELEC6027). It looks into 2 aspects of processor design - condition codes and conditional branches. The report begins with a overview of what should be achieved by the condition codes and branches. Three case studies of some simple architectures are then discussed, looking into how they work and includes code snippets. Some honourable mentions of other architectures are discussed also. The report then finishes with a comparison of the case studies and discusses the conclusions that can be drawn from this research The authors’ recommendation for the implementation is also given in this section.

One or more sections covering your research topic. Here you should include appropriate figures and code snippets to illustrate your discussion. Ensure that all figures and code snippets are properly explained in your text. Where text, figures or code snippets are copied from another source, the source must be clearly acknowledged. Copied text must be surrounded by quotation marks ”...” to show clearly that it is copied.

2 Operation

What is the point in flags and branches. Why do we need them.

3 Case studies

Three case studies are discussed here. For each architecture studied, there is a description of the flags implemented (if any) and instructions used for conditional branches. Also, a C code snippet, seen in listing 1, is converted to assembler for each architecture to see how the architectures compare.

Listing 1: C Code

```c
uint16_t a = 0;
for(uint16_t i = 0; i < 10; i++)
{
        a = a + i;
}
```

Flag	Shorthand	Explanation
Carry	C	
Overflow	V	
Zero	Z	Set if the ALU result
Negative	N	Set if the ALU result is less than 0, i.e. bit 7 is set high.

Table 1: Explanation of the flags in the ARM Cortex M0

3.1 ARM Cortex M0

The ARM Cortex M0 is a 32-bit RISC architecture that implements the Thumb/Thumb2 instruction set [2]. It implements 19 instructions in total, one of which is the ‘conditional branch’ instruction.

3.1.1 Flags

The Cortex M0 uses 4 flags; Carry, oVerflow, Zero and Negative [1]. The flags are stored in a specific register, the Application Program Status Register (APSR).

What triggers the flags. CMP CMN, TEQ, TST

3.1.2 Conditional Instructions

3.2 Intel 8086

3.3 MIPS

4 Other Architectures

4.1 DEC Alpha

4.2 AVR

Even though the AVR is a microcontroller core, it still posed interesting reading.

T Flag

5 Conclusion

Here I would like you to discuss how the issues raised in your report will affect your processor design. Where you have seen different processors opting for different solutions to the same problem you should discuss their relative merits in the context of your design.

References

[1] ARM Holdings. Arm information centre - updates to the alu status flags, 2014.

[2] ARM Holdings. Cortex m0 processor- arm, 2014.

Bibliography

[1] Leslie Lamport, *LaTeX: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.