Project #1 Report Template

1. SCOPE

This template gives an idea of performing instruction level analysis of a C function.

2. CODE

a. C Function

The C code presented is the version used when compiling for size.

```
Line#  C Code
    1  #include <stdlib.h>
    2  int
    3  _DEFUN (abs, (i), int i)
    4   {
    5     return (i < 0) ? -i : i;
    6   }
```

b. Assembly Function

Assembly code is compiled from the above C code using the –O3 optimization level.

```
Line# Assembly Code
      abs:
        @ args = 0, pretend = 0, frame = 0
        @ frame_needed = 0, uses_anonymous_args = 0
        @ link register save eliminated.
1       cmp r0, #0
2       it lt
3       rsblt r0, r0, #0
4       bx lr
```

3. Code Analysis

a. General Analysis

For each instruction, the following is presented:

1) Instruction
2) Register states prior to execution
3) Register states after execution
4) State of all condition codes (N,Z,V,C) after execution
5) What does the instruction do to help implement the C function?

b. Line-by-Line Analysis

i. Line #1

```
cmp    r0, #0
```

Compare r0 with 0.

This instruction performs the check at line 5 of the C function to determine if 'i' is greater than or less than zero.

Register	Before Instruction	After Instruction
r0		
r1		
r2		
r3		
ip (r12)		
lr (r14)		
pc (r15)		

CPSR bit	Value	Rationale
N	0	r0 cannot be negative
Z	0/1	if r0 = 0 then 1, else 0
V	0	r0 cannot be negative
C	1	r0 cannot be negative

4. Timing Analysis

Line	Instruction	Notes	Clk Cycles
1	cmp r0, #0		
2	it lt		
3	rsblt r0, r0, #0		
4	bx lr		

5. Big Picture