ELEC6027: VLSI Design Project
Part 1: Microprocessor Research
Topic: Subroutines

Ashley Robinson
Team: R4
Course Tutor: Mr B. Iain McNally

February 10, 2014

Contents

1 Introduction

2 Research

2.1 Subroutine Context Save

2.2 Operation of Stack Frames

2.2.1 8086

The assembler held in listing 1 and 2 is written for the Intel 8086 microprocessor. A basic example of how stack frames are built to pass parameters to and from a subroutine. The main program in listing 1 loads two immediate values into registers then begins building a stack frame by pushing them to the stack. Calling the procedure to act upon the arguments passed via the stack and finally destroying the stack frame by popping data, including any return arguments, into registers.

Listing 1: 8086Caller.asm

```
main:                   ; Main loop
      MOV   ax,42       ; Load arg1
      MOV   bx,69       ; Load arg2
      PUSH  ax          ; Push arg1 to stack
      PUSH  bx          ; Push arg2 to stack
      CALL  adder       ; Call the subroutine
      POP   ax          ; Dummy pop from arg2 spot
      POP   ax          ; Result pop from arg1 spot
      JMP   main
```

When the subroutine, in listing 2, is called the return address is pushed onto by using the *call* instruction. This will be

Listing 2: 8086Callee.asm

```
adder PROC              ; Subroutine
      PUSH  bp          ; Push base ptr to stack
      MOV   bp,sp       ; Set base ptr to stack ptr
      ADD   bp,4        ; Move to arg2 in stack
      MOV   ax,[bp]     ; Load into working reg
      ADD   bp,2        ; Move to arg1 in stack
      ADD   ax,[bp]     ; Add to contents of working reg
      MOV   [bp],ax     ; Replace arg1 with result
      POP   bp          ; Restore base ptr
      RET
adder ENDP
```

This code was tested upon an 8086 emulator [3]. The emulator provides a complete overview of the flow of data within the processor, including the stack.

2.2.2 ARM7TDMI using Arm Thumb

The ARM7TDMI is a 32-bit RISC microprocessor with an emphasis on low-power design and pipelining for high throughput [1]. It has two instruction sets one of which is Arm Thumb, a low density 16-bit subset of the ARM assembly language [2]. A user selectable flag is set to switch between instruction sets therefore drawing on each sets advantages.

This architecture does not have built-in support for calling subroutines using the stack. When the branch instruction is used, as seen in listing 3, the program counter is overwritten with the address of the corresponding label. The address of the next line of code, which should be returned to after the subroutine, is placed into the link register. Calling conventions suggests leaving this register untouched and simply moving the data back into the program counter on a return.

Listing 3: ArmCaller.asm

```
main  MOV   r0,#42    ; Load arg1
      MOV   r1,#69    ; Load arg2
      PUSH  r0        ; Push arg1 to stack
      PUSH  r1        ; Push arg2 to stack
      BL    adder     ; Branch to subroutine
      POP   r0        ; This line is held in the link register
      POP   r0        ; Result pop from arg1 spot
      BL    main
```

In this case the link register is pushed onto the stack from the subroutine therefore requiring the subroutine to pop the value into the program counter in order to return. Listing 4 holds the subroutine and handles placing the return address on the stack. Relative addressing on the stack is required to draw the two arguments out and replace the first with the output of the function.

Listing 4: ArmCallee.asm

```
adder PUSH  lr              ; Link register holds return address
      LDR   r0, [sp, #12]   ; Get arg1 off stack
      LDR   r1, [sp, #8]    ; Get arg2 off stack
      ADD   r0,r1           ; Do the add
      STR   [sp,#12], r0    ; Replace arg1 on the stack
      POP   pc              ; Restore program counter and return
```

3 Conclusion

References

[1] ARM Holdings plc. Arm7tdmi data sheet. `http://www.ndsretro.com/download/ARM7TDMI.pdf`, Aug 1995. Online. Acessed Feb 2014.

[2] ARM Holdings plc. Thumb instruction set quick reference card. `http://www.eng.auburn.edu/~nelson/courses/elec5260_6260/Thumb%20Instructions.pdf`, Oct 2003. Online. Acessed Feb 2014.

[3] Daniel B. Sedory, Randall Hyde, Eric Isaacson, Barry Allyn, Tomasz Grysztar, Saul Coval, Bob Brodt, Jordan Russell, and Jeremy Gordonii. emu8086. `http://www.emu8086.com/`, 2013. Online. Acessed Feb 2014.

Bibliography

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