

Tyler Ovenden

Prelab for lab #5

ESE 280

Note: I had a lot of issues regarding generating lss files; only for the conditional\_input task, the lss file contained the entire code. For the others I was unable to get the entire code to show up in the lss file, I attached both the lss files & assembly code to show what wouldn't show up. I tried to determine the problem by isolating various loops & parts of the code & running the code in a new project & all of these attempts didn't change anything.

AVRASM ver. 2.2.7 C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin  
 \pb\_bounce\_count\_bin\main.asm Thu Oct 08 14:13:48 2020

C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin\pb\_bounce\_count\_bin  
 \main.asm(8): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel  
 \ATmega\_DFP\1.3.300\avrasm\inc\m4809def.inc'

```

                                ; pb_bounce_count_bin.asm
                                ;
                                ; Created: 10/4/2020 2:03:45 PM
                                ; Author : tyler
                                ;
                                .list
                                ; Replace with your application code
                                start:
                                ; configure I/O ports
000000 ef0f                    ldi r16, 0xFF ;load r16 with all 1s
000001 b90c                    out VPORTD_DIR, r16 ;PORTD - all pins configured
                                as outputs
000002 e000                    ldi r16, 0x00 ; load r16 with all 0s
000003 2f30                    mov r19, r16 ; load r19 to be a counter

```

#### RESOURCE USE INFORMATION

-----

##### Notice:

The register and instruction counts are symbol table hit counts, and hence implicitly used resources are not counted, eg, the 'lpm' instruction without operands implicitly uses r0 and z, none of which are counted.

x,y,z are separate entities in the symbol table and are counted separately from r26..r31 here.

.dseg memory usage only counts static data declared with .byte

##### "ATmega4809" register use summary:

```

x : 0 y : 0 z : 0 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0
r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
r13: 0 r14: 0 r15: 0 r16: 4 r17: 0 r18: 0 r19: 2 r20: 0
r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 0 r27: 0 r28: 0
r29: 0 r30: 0 r31: 0

```

Registers used: 2 out of 35 (5.7%)

##### "ATmega4809" instruction use summary:

```

.lds : 0 .sts : 0 adc : 0 add : 0 adiw : 0 and : 0
andi : 0 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0

```

```

brne : 0 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
brvs : 0 bset : 0 bst : 0 call : 0 cbi : 0 cbr : 0
clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
clt : 0 clv : 0 clz : 0 com : 0 cp : 0 cpc : 0
cpi : 0 cpse : 0 dec : 0 des : 0 eor : 0 fmul : 0
fmuls : 0 fmulsu : 0 icall : 0 ijmp : 0 in : 0 inc : 0
jmp : 0 ld : 0 ldd : 0 ldi : 2 lds : 0 lpm : 0
lsl : 0 lsr : 0 mov : 1 movw : 0 mul : 0 muls : 0
mulsu : 0 neg : 0 nop : 0 or : 0 ori : 0 out : 2
pop : 0 push : 0 rcall : 0 ret : 0 reti : 0 rjmp : 0
rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
sev : 0 sez : 0 sleep : 0 spm : 0 st : 0 std : 0
sts : 0 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0

```

Instructions used: 3 out of 114 (2.6%)

"ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00000a	10	0	10	49152	0.0%
[.dseg]	0x002800	0x002800	0	0	0	6144	0.0%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

Assembly complete, 0 errors, 0 warnings

```

;
; pb_bounce_count_bin.asm
;
; Created: 10/4/2020 2:03:45 PM
; Author : tyler
;
.nolist
.include "m4809def.inc"
.list
; Replace with your application code
start:
; configure I/O ports
    ldi r16, 0xFF ;load r16 with all 1s
    out VPORTD_DIR, r16 ;PORTD - all pins configured as outputs
    ldi r16, 0x00 ; load r16 with all 0s
    mov r19, r16 ; load r19 to be a counter
    out VPORTE_DIR, r19 ; PORTE - all pins configured as inputs

;main_loop
main_loop:
    sbis VPORTE_IN, 0 ;skip if PORTDE0 == 1
    rjmp main_loop ;if PORTDE0 is 0 calls the loop again & will repeat until
    PORTDE0 is 1
    rjmp check_one ;if PORTDE0 is 1 calls check_one loop

check_one:
    sbic VPORTE_IN, 0 ;skip if PORTDE == 0
    rjmp check_one ;if PORTDE0 is 1 calls the loop again & will repeat until
    PORTDE0 is 0
    rjmp count ;if PORTDE0 is 0 calls loop to increment the counter

count:
    cpi r19, 0xFF ;compares the counter to a full 8 bit counter
    breq reset ;if r19 = 0xFF calls loop to set counter to 0
    inc r19 ;else the counter is increased by one
    out VPORT_OUT, r19 ;VPORTD is set to display the counter
    rjmp main_loop ;calls the main loop again

reset:
    ldi r19, 0x00 ;clears the counter by setting the register to 0

    out VPORT_OUT, r19 ;VPORTD is set to display the counter
    rjmp main_loop ;calls the main loop again

```

```

;
; pb_sfwe_debounce_count_bin.asm
;
; Created: 10/4/2020 3:50:51 PM
; Author : tyler
;

.nolist
.include "m4809def.inc"
.list
; Replace with your application code
start:
; configure I/O ports
    ldi r17, 0xFF          ;load r16 with all 1s
    out VPORTD_DIR, r17    ;PORTD - all pins configured as outputs
    ldi r17, 0x00          ; load r16 with all 0s
    out VPORTE_DIR, r17    ; PORTE - all pins configured as inputs
    ldi r19, 0x00          ; load r19 to be a counter

main_loop:
    sbis VPORTE_IN, 0      ;skip if PORTDE0 == 1
    rjmp main_loop        ;calls main_loop until PORTDE0 ==1
    rjmp first_one        ;calls first_one loop

first_one:
    ldi r16, 100;          ;loads r16 for using in delay subroutine
    sbic VPORTE_IN, 0      ;skip if PORTDE0 == 0
    rjmp first_one        ;if PORTDE0 is 1, reruns loop until it's 0
    rcall var_delay        ;calls var_delay subroutine for debouncing
    sbic VPORTE_IN, 0      ;checks if PORTDE0 is still 0
    rjmp main_loop        ;if it is now 1, calls first loop again
    rjmp count            ;PORTDE0 is so calls count loop

count:
    cpi r19, 0xFF          ;compares the counter to a full 8 bit counter
    breq reset            ;if r19 = 0xFF calls loop to set counter to 0
    inc r19               ;else the counter is increased by one
    out VPORT_OUT, r19     ;VPORTD is set to display the counter
    rjmp check_one        ;now calls check_one loop

reset:
    ldi r19, 0x00          ;clears the counter by setting the register to 0

    out VPORT_OUT, r19     ;VPORTD is set to display the counter
    rjmp check_one        ;calls check_one loop

check_one:
    ldi r16, 100;          ;loads r16 for using in delay subroutine
    sbis VPORTE_IN, 0      ;checks if VPORTE0 is 1
    rjmp check_one        ;keep restarting loop until PORTDE is 1
    rcall var_delay        ;calls var_delay subroutine for debouncing

```

---

```
sbis VPORTE_IN, 0          ;checks if VPORTDE0 is still 1
rjmp check_one             ;restarts loop if it is 0
rjmp main_loop             ;restarts main_loop

var_delay:
outer_loop:
ldi r17, 110               ;loads r17 with 110
inner_loop:
dec r17                   ;decreases r17
brne inner_loop           ;branches to start of inner_loop if not equal
dec r16                   ;decreases 16
brne outer_loop           ;branches to outer_loop if not equal
ret                       ;ends subroutine
```

AVRASM ver. 2.2.7 C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin  
 \unconditional\_input\main.asm Thu Oct 08 13:51:21 2020

C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin\unconditional\_input  
 \main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel  
 \ATmega\_DFP\1.3.300\avrasm\inc\m4809def.inc'  
 C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin\unconditional\_input  
 \main.asm(9): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel  
 \ATmega\_DFP\1.3.300\avrasm\inc\m4809def.inc'

```

; unconditional_input.asm
;
; Created: 10/4/2020 3:35:17 PM
; Author : tyler
;

.list
; Replace with your application code
start:
; configure I/O ports
000000 ef0f      ldi r16, 0xFF ;load r16 with all 1s
000001 b90c      out VPORTD_DIR, r16 ;PORTD - all pins configured
               as outputs
000002 e000      ldi r16, 0x00 ; load r16 with all 0s
000003 b900      out VPORTA_DIR, r16 ; PORTA - all pins configured
               as inputs

main_loop:
000004 b102      in r16, VPORTA_IN ;load r16 with input from the
               switch
000005 b90d      out VPORTD_OUT, r16 ;display inputs from switch
               onto 7 segment display
000006 cffd      rjmp main_loop ;repeat the loop

```

#### RESOURCE USE INFORMATION

-----

#### Notice:

The register and instruction counts are symbol table hit counts, and hence implicitly used resources are not counted, eg, the 'lpm' instruction without operands implicitly uses r0 and z, none of which are counted.

x,y,z are separate entities in the symbol table and are counted separately from r26..r31 here.

.dseg memory usage only counts static data declared with .byte

"ATmega4809" register use summary:

x : 0 y : 0 z : 0 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0



```

r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
r13: 0 r14: 0 r15: 0 r16: 6 r17: 0 r18: 0 r19: 0 r20: 0
r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 0 r27: 0 r28: 0
r29: 0 r30: 0 r31: 0

```

Registers used: 1 out of 35 (2.9%)

"ATmega4809" instruction use summary:

```

.lds : 0 .sts : 0 adc : 0 add : 0 adiw : 0 and : 0
andi : 0 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
brne : 0 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
brvs : 0 bset : 0 bst : 0 call : 0 cbi : 0 cbr : 0
clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
clt : 0 clv : 0 clz : 0 com : 0 cp : 0 cpc : 0
cpi : 0 cpse : 0 dec : 0 des : 0 eor : 0 fmul : 0
fmuls : 0 fmulsu: 0 icall : 0 ijmp : 0 in : 1 inc : 0
jmp : 0 ld : 0 ldd : 0 ldi : 2 lds : 0 lpm : 0
lsl : 0 lsr : 0 mov : 0 movw : 0 mul : 0 muls : 0
mulsu : 0 neg : 0 nop : 0 or : 0 ori : 0 out : 3
pop : 0 push : 0 rcall : 0 ret : 0 reti : 0 rjmp : 1
rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
sev : 0 sez : 0 sleep : 0 spm : 0 st : 0 std : 0
sts : 0 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0

```

Instructions used: 4 out of 114 (3.5%)

"ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00000e	14	0	14	49152	0.0%
[.dseg]	0x002800	0x002800	0	0	0	6144	0.0%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

Assembly complete, 0 errors, 0 warnings

AVRASM ver. 2.2.7 C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin  
 \conditional\_input\_sftwe\main.asm Thu Oct 08 14:23:23 2020

C:\Users\tyler\Documents\Atmel Studio\7.0\pb\_bounce\_count\_bin\conditional\_input\_sftwe  
 \main.asm(8): Including file 'C:/Program Files (x86)\Atmel\Studio\7.0\Packs\atmel  
 \ATmega\_DFP\1.3.300\avrasm\inc\m4809def.inc'

```

                                ; conditional_input_sftwe.asm
                                ;
                                ; Created: 10/7/2020 5:48:15 PM
                                ; Author : tyler
                                ;
                                .list

                                ; Replace with your application code
                                start:
                                ; configure I/O ports
000000 ef1f                    ldi r17, 0xFF ;load r16 with all 1s
000001 b91c                    out VPORTD_DIR, r17 ;PORTD - all pins configured
                                as outputs
000002 e010                    ldi r17, 0x00 ;load r16 with all 0s
000003 bb10                    out VPORTE_DIR, r17 ;PORTA - all pins configured
                                as inputs
000004 b910                    out VPORTA_DIR, r17 ;PORTA - all pins configured
                                as inputs
                                mov r19, r17 ;load r19 to be a counter

                                main_loop:
                                sbis VPORTE_IN, 2 ;checks if PORTE2 (switch button) is 1
                                rjmp main_loop ;restarts loop when switch button

```

#### RESOURCE USE INFORMATION

-----

#### Notice:

The register and instruction counts are symbol table hit counts, and hence implicitly used resources are not counted, eg, the 'lpm' instruction without operands implicitly uses r0 and z, none of which are counted.

x,y,z are separate entities in the symbol table and are counted separately from r26..r31 here.

.dseg memory usage only counts static data declared with .byte

"ATmega4809" register use summary:

```

x : 0 y : 0 z : 0 r0 : 0 r1 : 0 r2 : 0 r3 : 0 r4 : 0
r5 : 0 r6 : 0 r7 : 0 r8 : 0 r9 : 0 r10: 0 r11: 0 r12: 0
r13: 0 r14: 0 r15: 0 r16: 1 r17: 5 r18: 0 r19: 0 r20: 0

```

r21: 0 r22: 0 r23: 0 r24: 0 r25: 0 r26: 0 r27: 0 r28: 0

r29: 0 r30: 0 r31: 0

Registers used: 2 out of 35 (5.7%)

"ATmega4809" instruction use summary:

```
.lds : 0 .sts : 0 adc : 0 add : 0 adiw : 0 and : 0
andi : 0 asr : 0 bclr : 0 bld : 0 brbc : 0 brbs : 0
brcc : 0 brcs : 0 break : 0 breq : 0 brge : 0 brhc : 0
brhs : 0 brid : 0 brie : 0 brlo : 0 brlt : 0 brmi : 0
brne : 0 brpl : 0 brsh : 0 brtc : 0 brts : 0 brvc : 0
brvs : 0 bset : 0 bst : 0 call : 0 cbi : 0 cbr : 0
clc : 0 clh : 0 cli : 0 cln : 0 clr : 0 cls : 0
clt : 0 clv : 0 clz : 0 com : 0 cp : 0 cpc : 0
cpi : 0 cpse : 0 dec : 0 des : 0 eor : 0 fmul : 0
fmuls : 0 fmulsu : 0 icall : 0 ijmp : 0 in : 0 inc : 0
jmp : 0 ld : 0 ldd : 0 ldi : 2 lds : 0 lpm : 0
lsl : 0 lsr : 0 mov : 0 movw : 0 mul : 0 muls : 0
mulsu : 0 neg : 0 nop : 0 or : 0 ori : 0 out : 4
pop : 0 push : 0 rcall : 0 ret : 0 reti : 0 rjmp : 0
rol : 0 ror : 0 sbc : 0 sbci : 0 sbi : 0 sbic : 0
sbis : 0 sbiw : 0 sbr : 0 sbrc : 0 sbrs : 0 sec : 0
seh : 0 sei : 0 sen : 0 ser : 0 ses : 0 set : 0
sev : 0 sez : 0 sleep : 0 spm : 0 st : 0 std : 0
sts : 0 sub : 0 subi : 0 swap : 0 tst : 0 wdr : 0
```

Instructions used: 2 out of 114 (1.8%)

"ATmega4809" memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00000c	12	0	12	49152	0.0%
[.dseg]	0x002800	0x002800	0	0	0	6144	0.0%
[.eseg]	0x000000	0x000000	0	0	0	256	0.0%

Assembly complete, 0 errors, 0 warnings

```

;
; conditional_input_sftwe.asm
;
; Created: 10/7/2020 5:48:15 PM
; Author : tyler
;
.nolist
.include "m4809def.inc"
.list

; Replace with your application code
start:
; configure I/O ports
    ldi r17, 0xFF ;load r17 with all 1s
    out VPORTD_DIR, r17 ;PORTD - all pins configured as outputs
    ldi r17, 0x00 ;load r17 with all 0s
    out VPORTE_DIR, r17 ;PORTA - all pins configured as inputs
    out VPORTA_DIR, r17 ;PORTA - all pins configured as inputs
    mov r19, r17 ;load r19 to be a counter

main_loop:
    sbis VPORTE_IN, 2 ;checks if PORTE2 (switch button) is 1
    rjmp main_loop ;restarts loop when switch button is 0
    rjmp first_one ;calls first_one loop if PORTE2 is 1

first_one:
    ldi r16, 100; ;sets register to 100 for delay
    rcall var_delay ;calls delay subroutine for debouncing
    sbis VPORTE_IN, 2 ;checks if PORTE2 (switch button) is still one 1
    rjmp main_loop ;if it is no longer 1, restarts main_loop
    rjmp display ;calls display function

display:
    sbi VPORTE_IN , 1 ;sets flip flop output to 1
    cbi VPORTE_IN , 0 ;sets clear to 0
    in r16, VPORTA_IN ;loads switch values into register
    out VPORTD_OUT, r16 ;sets display to switch values
    rjmp check_zero ;calls check_zero loop

check_zero:
    ldi r16, 100; ;sets register to 100 for delay
    sbis VPORTE_IN, 2 ;checks if push button is released
    rjmp check_zero ;if push button still down restarts loop
    rcall var_delay ;calls delay subroutine for debouncing
    sbis VPORTE_IN, 2 ;checks if push button is still released
    rjmp check_zero ;if push button no longer released restart loop
    rjmp reset ;calls loop to reset & clear the display

```

```

reset:
    ldi r16, 0x00          ;clear a register
    out VPORTD_OUT, r16   ;clears the display
    cbi VPORTE_IN , 1     ;sets flip flop output to 0
    sbi VPORTE_IN , 0     ;sets clear to 0
    rjmp main_loop        ;return to main_loop

var_delay:
    outer_loop:
        ldi r17, 110      ;loads r17 with 110
        inner_loop:
            dec r17        ;decreases r17
            brne inner_loop ;branches to start of inner_loop if not equal
            dec r16        ;decreases 16
            brne outer_loop ;branches to outer_loop if not equal
            ret            ;ends subroutine

```

