University of Colorado at Boulder
Department of Electrical and Computer Engineering
ECEN 2120 – Computers as Components

Quiz II

February 2008

STUDENT ID

- TIME ALLOWED = 50 minutes
- THIS IS A CLOSED BOOK EXAM
- NO CALCULATOR IS ALLOWED

Q1	Q2

TOTAL	

Q1 (40pts) Please translate the following C code into MC68K assembly. Array a is stored in the main memory at starting address 0X80000, and array b is stored in the main memory at starting address 0XA0000.

```
int a[100];
int b[100];

for(i = 0; i < 100; i++) {
    b[i] = a[99-i];
}
```

```
        move.l #$800000, a0         ;a
        move.l #$A0000, a1          ;b
        move.l #0, d0

Loop:
        cmp.l #99 d0
        beq End
        move.l #99, d1
        sub.l d0, d1
        move.l (a0, d1), (a1, d0)
        add.l #1, d0
        bra Loop
End:
        ret
```

Q2 (60pts) Given the following recursive C and the corresponding assembly program, please fill in the stack with exact values for (1). max_search(a, 2, 3) is invoked (2) max_search(a, 1, 3) returns. Hint: the following program determines the largest data item stored in array a.

```
#define N 3
int max_search(int a[], int start, int end);

int main() {

  int a[N];
  int i, max_value;

  for(i = 0; i < N; i++) {
     a[i] = i;
  }

  max_value = max_search(a, 0, N);

  return max_value;

}
```

```
int max_search(int a[], int start, int end) {
  int max_r;
  if (start == (end-1)) {
     return a[start];
  }else {
     max_r = max_search(a, start+1, end);
     if (a[start] > max_r) {
       return a[start];
     }else {
       return max_r;
     }
}}
```

```
==========================================================================
;       Microtec(R) 68K C Compiler 5.3
        TTL     .\re_max2.c
        OPT
        NOABSPCADD,E,NOPCR,P=68000,CASE
        NAME    re_max2
        SECTION code,,C
        XDEF    _main
_main:
        link    a6,#-16
        movem.l d2/d3,-(sp)
        moveq   #0,d2
        bra.s   L2
L1:
        move.l  d2,d0
        lsl.l   #2,d0
        move.l  d2,-16(a6,d0.l)
        addq.l  #1,d2
L2:
        moveq   #3,d0
        cmp.l   d2,d0
        bgt.s   L1
        moveq   #3,d0
        move.l  d0,-(sp)
        move.l  #0,-(sp)
        pea     -16(a6) ;push address of a[] to the stack
        jsr     _max_search
        move.l  d0,d3
        movem.l -24(a6),d2/d3
        unlk    a6
        rts
;        code: 56 bytes   stack: 16 bytes
        XDEF    _max_search
_max_search:
        link    a6,#0
        movem.l a2/d2/d3/d4,-(sp)
        movea.l 8(a6),a2
        move.l  12(a6),d2
        move.l  16(sp),d4
        move.l  d4,d0
        subq.l  #1,d0
        cmp.l   d2,d0
        bne.s   L5
        move.l  d2,d0
        lsl.l   #2,d0
        move.l  (a2,d0.l),d0
        bra     DONE

L5:  ; if start != (end -1)
        move.l  d4,-(sp)
        move.l  d2,d0
        addq.l  #1,d0
        move.l  d0,-(sp)
        move.l  a2,-(sp)
        bsr.b   _max_search
        add.l   #12,sp
        move.l  d0,d3
        move.l  d2,d0
        lsl.l   #2,d0
        cmp.l   (a2,d0.l),d3
        bge.s   L7
        move.l  d2,d0
        lsl.l   #2,d0
        bra     DONE

L7:  ; if  a[start] < max_r
        move.l  d3,d0
DONE:
        movem.l (sp)+,a2/d2/d3/d4
        unlk    a6
        rts
;       code: 88 bytes   stack: 0 bytes
        END
```

(1). Right after max_search(a, 2, 3) is invoked.

Ret pc
&a
2
3
D4
D3
D2
A2
A6
Ret pc
&a
1
3
D4
D3
D2
A2
OldA6
Ret pc
&a
0
3
D3
D2
0
1
2
3
Old A6

(2) Right after Max_search(a, 1, 3) returns

&a
1
3
D4
D3
D2
A2
OldA6
Ret pc
&a
0
3
D3
D2
0
1
2
3
Old A6