Homework Lecture 6

5.6

Write the binary form for the micro instructions shown below. Use don't care ('-') for fields that are not used.

```
60: R[temp0] ← ORN(R[0],R[temp0]]; IF Z THEN GOTO 64
61: R[rd] ← INC(R[rs1])
```

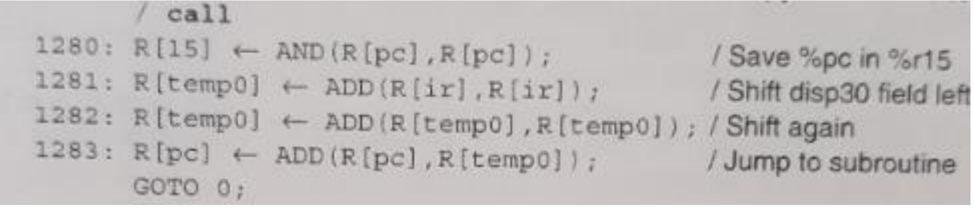
5.7

Three binary words are shown below, each of which can be interpreted as a micro instruction. Write the mnemonic version using micro assembly language.

A	Amux	B	Bmux	C	Cmux	RD	WR	ALU	COND	JUMP ADDR
100101	0	000000	0	100001	0	0	0	1100	000	00000000000
000000	1	100001	0	100001	0	0	0	1000	110	11100000001
000000	1	000000	1	100001	0	0	0	1000	101	11100010010

5.8

Page 165



```
      / call
1280: R[15] ← AND(R[pc],R[pc]);            / Save %pc in %r15
1281: R[temp0] ← ADD(R[ir],R[ir]);         / Shift disp30 field left
1282: R[temp0] ← ADD(R[temp0],R[temp0]);   / Shift again
1283: R[pc] ← ADD(R[pc],R[temp0]);         / Jump to subroutine
      GOTO 0;
```

The execution of the call instruction is shown above. Find an alternative with only 3 micro instructions instead of 4. Hint: use the ALU operation LSHIFT2.

5.9 a)

The organization of the microstrore is:

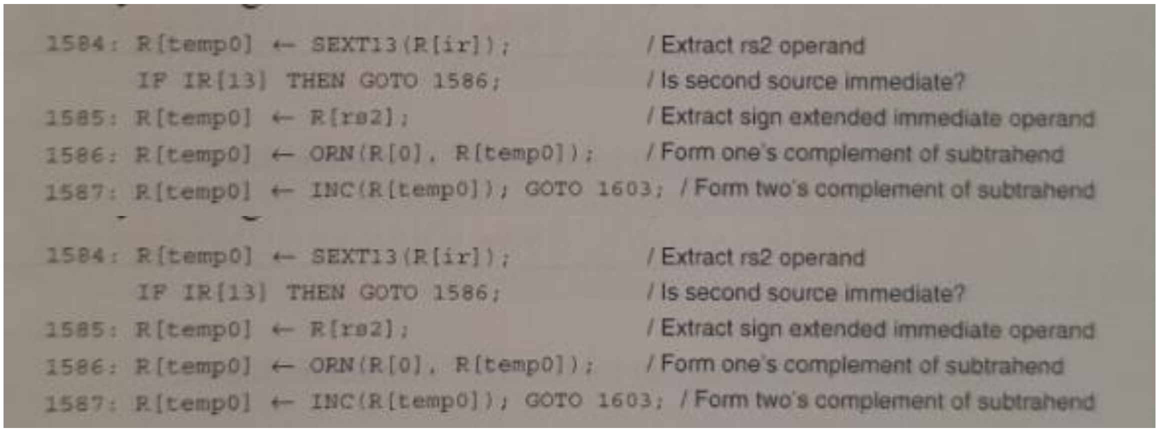
Micro address 0: on this address an instruction is read from main memory (address in main memory of that instruction is in PC), and then goes to micro address is 1.

Micro address 1: the instruction is decoded and then goes to the start address in the micro store of the execution phase of the instruction.

Micro address 2047: Many instructions will jump at the end of the execution phase to this address. At this address the PC is incremented with 4 ("points to the next instruction") and then goes to micro address 0.

The arithmetic instruction SUBCC is added to the instruction set with bit pattern for OP3 field: 001100 [1]
subcc %r1, %r2, %r3 ;meaning %r3 ←%r1-%r2 (short hand for R[%r3] ←R[%r1]-R[%r2])
Check that the start address of the execution phase of the call instruction is decimal 1584.

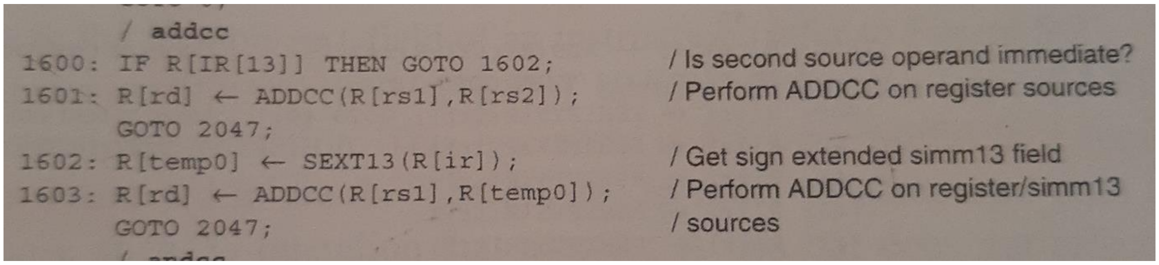
[1] You check with the ARCTools that this is not the correct value for field OP3. The correct value is 010101 (see also bottom of page 117). For this assignment the incorrect value is used.



```
1584: R[temp0] ← SEXT13(R[ir]);              / Extract rs2 operand
      IF IR[13] THEN GOTO 1586;               / Is second source immediate?
1585: R[temp0] ← R[rs2];                      / Extract sign extended immediate operand
1586: R[temp0] ← ORN(R[0], R[temp0]);         / Form one's complement of subtrahend
1587: R[temp0] ← INC(R[temp0]); GOTO 1603;    / Form two's complement of subtrahend
```

```
1584: R[temp0] ← SEXT13(R[ir]);              / Extract rs2 operand
      IF IR[13] THEN GOTO 1586;               / Is second source immediate?
1585: R[temp0] ← R[rs2];                      / Extract sign extended immediate operand
1586: R[temp0] ← ORN(R[0], R[temp0]);         / Form one's complement of subtrahend
1587: R[temp0] ← INC(R[temp0]); GOTO 1603;    / Form two's complement of subtrahend
```

The example is on page 171.



```
      / addcc
1600: IF R[IR[13]] THEN GOTO 1602;            / Is second source operand immediate?
1601: R[rd] ← ADDCC(R[rs1],R[rs2]);           / Perform ADDCC on register sources
      GOTO 2047;
1602: R[temp0] ← SEXT13(R[ir]);               / Get sign extended simm13 field
1603: R[rd] ← ADDCC(R[rs1],R[temp0]);         / Perform ADDCC on register/simm13
      GOTO 2047;                              / sources
      / andcc
```

Write the micro addresses that are visited during execution of the microinstruction of the SUBCC instruction (including fetch, decode, and update PC).

5.14

You have to add a microprogram for the instruction XORCC (XOR operation and condition code is set)
a) The ARCTools support the XORCC, what is the start address of this instruction in the microstore?
Tip: assemble the following program and find the values for op and op3 in the machine code:

```
.begin
.org 0
        xorcc %r1, %r2, %r3
        halt
.end
```

b) Give the RTL for the execution phase of XORCC
Note: you may assume that in the microstore the addresses 21 until 30 are not used.
c) Give the micro instructions for the execution phase of XORCC.

Exercise 1

What is the ARC assembly instruction for the machine code (hexadecimal): 82806004

Exercise 2

Given is an array `arr1` that contains numbers (size of a number is 32 bits). The end of the array is indicated with the integer value 0. Write an assembly program that count the number of occurrences of the decimal number 2 is this array. The number of occurrences must be stored in %r10. **Add comment!**

Simulate the program for different arrays with the ARCTools

```
      .begin
      .org 0
      sethi arr1, %r1
      srl %r1,10,%r1    ! the address of arr1 in  register %r1


< your program >



finished:   halt
      .org 200
arr1: 5,4,3,6,2,3,4,2,1,2,0 ! end of row is indicated with 0
      .end
```