Instructions**: Be sure your name is on your Scantron sheet and that the Exam ID matches the Exam ID above. Do not put your ID number on the Scantron sheet. Answer all questions on your Scantron sheet.**

**It is my policy to give a failing grade in the course to anyone who gives or receives unauthorized aid on any exam or quiz.**

---

Note: **Throughout this exam, assume the following ARC registers contain the following values. In general, a change to a register in one question does not affect its contents for any other question, but there are explicit exceptions to this, so read each question carfully. For example, if the answer to one question is "Register 3 changes to 0", you are to assume that Register 3 still contains FFFFFFFD when you work on the next question. %pc and %ir refer to the program counter (%r32) and the instruction register (%37) throughout this exam. All values are hexadecimal, except %ir, which is in binary with 'x' indicating "unspecified". Ignore the %ir bits shown for questions like numbers 1 and 2, where a specific ISA instruction is mentioned in the question.**

| | | | | | |
|---|---|---|---|---|---|
| **%r1:** | **FFFFFFFF** | **%r5:** | **00000504** | **%pc:** | **00000400** |
| **%r2:** | **FFFFFFFE** | **%r6:** | **00000604** | **%ir:** | **xx 00001 xxxxxx 00010 x 11111111 11110** |
| **%r3:** | **FFFFFFFD** | **%r7:** | **00000704** | | |
| **%r4:** | **FFFFFFFC** | **%r8:** | **00000804** | | |

**Also, assume memory locations have been initialized as follows before each question:**

| **Address** | **Contents** | | | |
|---|---|---|---|---|
| **400** | **86804002** | **86806002** | **88902500** | **c2016004** |
| **500** | **00000050** | **00000051** | **00000052** | **00000053** |
| **600** | **00000060** | **00000061** | **00000062** | **00000063** |
| **700** | **00000070** | **00000071** | **00000072** | **00000073** |
| **800** | **00000080** | **00000081** | **00000082** | **00000083** |

1. **What would be the effect of the instruction, "andcc %r0, %r0, %r3" when it is executed?**
   - **A) Register 3 would change to 0.**
   - **B) Register 3 would not change to 0.**
   - **C) Register 0 would change to 3.**
   - **D) Register 0 would explode.**
   - **E) Register 0 would implode.**
2. **What would be the effect of the instruction, "andcc %r3, %r3, %r3" when it is executed?**
   - **A) Register 3 would remain FFFFFFFD.**
   - **B) Register 3 would remain 0.**
   - **C) Register 3 would change to 3.**
   - **D) Register 3 would explode.**
   - **E) Register 3 would implode.**

**3. Assume the instruction in Question 2 was just executed. What will be the effect of the instruction, "addcc %r3, 1, %r3" when it is executed?**

**A) Register 3 would become FFFFFFFE.**
**B) Register 3 would change to 0.**
**C) Register 3 would remain 0.**
**D) Register 3 would remain FFFFFFFD.**
**E) Register 3 would change to 1.**

**If your answers to the first three questions were not "A," "A," and "A," read the note above before continuing with this exam!**

**4. What will register will change to what when the next instruction is fetched?**

**A) Register %pc will change to 86804002.**
**B) Register %ir will change to 86804002.**
**C) The condition codes will change from cvnz to zvnc.**
**D) Register %r1 will change to 00000001.**
**E) Register %r2 will change to 00000002.**

**5. Translate the instruction at location 408 to ARC assembly language.**

**A) .org 1024**
**B) addcc %r1, %r2, %r3**
**C) addcc %r1, 2, %r3**
**D) orcc %r0, 0x500, %r4**
**E) ld [%r5 + 4], %r1**

**6. What will be the effect of executing the instruction, "addcc %r1, 2, %r3"?**

**A) Register 1 will change to FFFFFF27**
**B) Register 2 will change to 00000002**
**C) Register 3 will change to FFFFFFFD (no change)**
**D) Register 3 will change to 00000001**
**E) Register 3 will change to FFFFFFFF**

**7. What will happen to the condition code bits as a result of executing the instruction, "addcc %r2, 2, %r3"?**

**A) C=0, V=0, N=0, Z=0**
**B) C=1, V=1, N=0, Z=1**
**C) C=0, V=1, N=0, Z=1**
**D) C=1, V=0, N=0, Z=1**
**E) C=0, V=0, N=1, Z=0**

**8. Translate the instruction at location 404 to ARC assembly language.**

**A) .org 1024**
**B) addcc %r1, %r2, %r3**
**C) addcc %r1, 2, %r3**
**D) orcc %r0, 0x500, %r4**
**E) ld [%r5 + 4], %r1**

**9. What would be the effect of executing the instruction, "ld [%r8 + 4], %r9"?**

**A) Register 9 changes to 00000080**
**B) Register 9 changes to 00000081**
**C) Register 9 changes to 00000082**

D) Register 9 changes to 00000083
E) Register 9 changes to 00000084

10. What can you do with tristate gates that you can't do with regular gates?
    A) Connect the outputs of two gates directly to each other.
    B) Connect inputs of two gates directly to each other.
    C) Connect the outputs of one gate to the input of another gate.
    D) Connect the input to one gate from the output of another gate.
    E) Create a short circuit.
11. Why are decoders often used in conjunction with tristate gates?
    A) You can build a decoder using only tristate buffers and inverters.
    B) You can build a tristate gate out of a decoder.
    C) You can use a decoder to be sure no more than one tristate gate is enabled at the same time.
    D) Tristate gates can store more information when used with a decoder.
    E) Decoders are more efficient than tristate gates unless they are used as flip-flops.
12. A single wire would connect to which input or output of all the flip-flops in a register?
    A) The Q inputs.
    B) The D outputs.
    C) The Q outputs.
    D) The D inputs.
    E) The clock inputs.
13. What does a branch instruction do?
    A) It sets the condition code bits to indicate the result of the branch.
    B) It tests the condition code bits to decide whether to change the contents of %ir or not.
    C) It tests the condition code bits to decide whether to change the contents of % pc or not.
    D) It loads %r15 with the address of the next instruction.
    E) It loads %r0 with the address of the next instruction.
14. What is the purpose of the cond field (%ir bits 25-28)?
    A) It tells what bits of the condition code to toggle.
    B) It tells what bits of the condition code to reset.
    C) It tells what bits of the condition code to set.
    D) It tells what bits of the condition code to test.
    E) It tells the condition of the electrical signals coming from memory.

**15. How many flip-flops are there inside the ARC ALU?**
- **A) 0**
- **B) 1**
- **C) 16**
- **D) 32**
- **E) 64**

**16. What is a lookup table?**
- **A) A wooden device for supporting a computer at a convenient height above the floor.**
- **B) A metal device for supporting a computer at a convenient height above the floor.**
- **C) A read only memory used inside the ALU.**
- **D) The registers and ALU, taken as a unit.**
- **E) The datapath and control unit, taken as a unit.**

**17. What is programmable about programmable logic devices?**
- **A) They use C, C++, or Java to compute the result of an ALU operation.**
- **B) They use Pascal, Fortran, or Cobol to compute the result of an ALU operation.**
- **C) They use assembly language to compute the result of an ALU operation.**
- **D) They use fuses to protect the result of an ALU operation.**
- **E) They use fuses to configure the connections among gates.**

**18. What would be the inputs and outputs of a 2048x41 Read Only Memory?**
- A) **2048 address wires in, and 41 data wires out.**
- **B) 11 address wires in, 41 data wires in, and 41 data wires out.**
- **C) 11 address wires in, and 41 data wires out.**
- **D) 41 address wires in, and 2048 data wires out.**
- **E) No inputs, 2048 address wires out, and 41 data wires out.**

**19. Where is the control store of a microprogrammed computer located?**
- **A) The control strore is located inside the ALU.**
- **B) The ALU is located inside the control store.**
- **C) The control store is located inside the registers.**
- **D) The control store is inside the control unit.**
- **E) Main Memory is inside the control store.**

**20. How many microinstructions are executed in one clock cycle?**
- **A) Exactly 0.**
- **B) Exactly 1.**
- **C) 0 or more.**
- **D) At least one.**
- **E) At least 2.**

**21. What register changes state every clock cycle?**
- **A) The Program Counter.**
- **B) The Instruction Register.**
- **C) Register 0.**
- **D) The Microcode Instruction Register.**

**E) The A Multiplexor.**

**22. What bus is used for all memory operations, and why?**

**A) The A Bus supplies the address to memory for both read and write operations.**
**B) The A Bus supplies the data to memory for both read and write operations.**
**C) The B Bus supplies the address to memory for both read and write operations.**
**D) The B Bus supplies the data to memory for both read and write operations.**
**E) The C bus gets the address from memory for both read and write operations.**

**23. What is a name for fetching and executing a single ISA instruction?**

**A) A minor cycle.**
**B) A major cycle.**
**C) A decode cycle**
**D) An execute cycle.**
**E) The Krebs cycle.**

**24. How many minor cycles are there in one major cycle of the microprogrammed version of the ARC?**

**A) Exactly 1 (fetch).**
**B) Exactly 2 (fetch and decode).**
**C) Exactly 3 (fetch, decode, and execute).**
**D) Exactly 4 (fetch, decode, execute, and store result)**
**E) At least 3.**

**The arguments to *do_minor()*, from left to right, are: The A Bus register, the B Bus register, the C Bus register, the ALU operation, and the Memory operation. Assume the bits given at the beginning of the exam are in the %ir for each of the following calls to *do_minor()*.**

**25. What does "do_minor( rs1, rs2, rd, addcc, nop )" do?**

**A) Add %r2 and %r30, putting the result in %r1.**
**B) Add %r1 and %r2, putting the result in %r30.**
**C) Add -2 to %r2, putting the result in %r1.**
**D) Add the contents of memory word 0x500 to %pc.**
**E) Load the contents of memory word 0x400 into %ir.**

**26. What does "do_minor( pc, r0, ir, and, read )" do?**

**A) Puts all zeros in the instruction register.**
**B) Adds whatever is in %r0 to the program counter.**
**C) Copies the pc into the ir.**
**D) Copies the ir into the pc.**
**E) Fetches the next ISA instruction.**

**27. What does the following sequence of calls to *do_minor()* do?**

**do_minor( ir, r0, temp0, sext13, nop )**
**do_minor( r2, temp0, temp0, add, nop )**
do_minor( temp0, r8, temp0, add, nop )

**A) It puts 0500 in %temp0**
**B) It puts 0600 in %temp0**
**C) It puts 0700 in %temp0**
**D) It puts 0800 in %temp0**
**E) It puts the first instruction in %temp0.**