

The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Fall 2012
Test 2
November 20, 2012

Name: _____

1. (1 point). Pipelining is a technique for improving the _____ of instruction execution.
2. (1 point) In _____ microcode, each control signal is represented by a bit in the microinstruction.
3. (1 point) In _____ microcode, each distinct microinstruction is encoded as a single signal that is fanned out to the control signals that are to be asserted.
4. (1 point) _____ RAM requires refreshing.
5. (1 point) The bit line _____ operation that causes the difference between access time and cycle time
6. (4 points) r2 contains a value of 235 in decimal. What is the decimal value of r1 after this instruction is executed?

shc r1, r2, 27

7. (3 points) r2 contains a value of -5179 in decimal. What is the decimal value of r1 after this instruction is executed?

shra r1, r2, 4

8. (2 points) Which memory is the largest? Select one.

(a) register (b) cache memory (c) main memory (d) disk memory

9-13. (20 points) What are the values of the following registers and memory address when the program executes “brzr r29, r4” for the fourth time? Answer in decimal.

9. r30: _____

10. r1: _____

11. r2: _____

12. r3: _____

13. r4: _____

```
data:      .org 200
num1:      .dc 8
num2:      .dc 9
result:    .dw 1
```

```
code:      .org 1000
           lar r30, again
           lar r29, done
           ld  r1, num1
           ld  r2, num2
           sub r3, r3, r3
           brzr r29, r1
           brzr r29, r2
           addi r4, r2, 0
again:     add  r3, r3, r1
           addi r4, r4, -1
           brzr r29, r4
           br  r30
done:     st  r3, result
           stop
```

14. (10 points) Identify all of the data dependencies in the following code.

```
add  r3, r5, r4
ld   r4, 28(r1)
add  r5, r4, r2
st   r5, 100(r3)
add  r6, r5, r8
brpl r29, r5
```

15. ____ (2 points) Which is **not** a register in SRC?
- 32 general purpose registers
 - a program counter
 - a cause register
 - an instruction register
16. ____ (2 points) Suppose that R[rc] contains 0x004B 0000. When does a branch not occur?
- $c3 < 2..0 > = 1$
 - $c3 < 2..0 > = 3$
 - $c3 < 2..0 > = 4$
 - $c3 < 2..0 > = 5$
17. ____ (2 points) This is 1-bus microarchitecture. We add new registers not found in the abstract RTN. Which is incorrect?
- A and C registers are needed to temporarily store two operands and the result when doing ALU operations.
 - MA and MD registers are used as interface registers to the memory system.
 - MA contains the address of the memory operand.
 - MD is used as a buffer for outgoing and incoming values.
 - None of the above
18. ____ (2 points) Which is incorrect in 1-bus SRC?
- MA only receives data from the bus and passes it on to the memory subsystem.
 - MD is bidirectional: it sends data to the memory subsystem upon a "write" and receives data from the memory system upon a "read."
 - The 4-bit register, n, stores the shift count, and in this design n is a counter that can be decremented.
 - The Cond logic employs the least significant 3 bits of the c3 field in the IR, which specify the condition to be tested in conditional branch instruction, as well as the bus contents to compute whether a given condition is true.
 - None of the above
19. ____ (2 points) If the maximum clock frequency is 625 MHz and we assume no safety margin, what is the minimum clock period?
- 1.6 ns
 - 800 ps
 - 1.6 ps
 - 800 ns
20. ____ (2 points) Which is the incorrect assumption in pipeline design?
- The instruction set is unchanged.
 - Two separate memories are needed, one for program and the other for data.
 - Virtually all pipelined designs require a 3-port register file to allow the reading of one operand and the writing of the other in a single clock cycle.
 - Pipelined designs replace the buses with direct connections between registers.
 - None of the above.
21. ____ (2 points) Which is incorrect explaining the global state of the pipelined SRC?
- Instruction memory is only accessed in Stage 1.
 - PC is only accessed in Stage 1.
 - General registers are read in Stage 2 and written in Stage 5.
 - Data memory is only accessed in Stage 4.
22. ____ (2 points) There are 9 ALU and 3 branch control signals. If we use a 4:16 decoder and a 2:4 decoder in vertical microcode, how many bits are saved compared to horizontal microcode?
- 6
 - 8
 - 7
 - 14

23. (10 points) Design a microcode sequence to implement the 1-bus SRC XOR instruction using the concrete RTN shown below.

address	100	101	102	700	701	702	703	704	705	706	707	708	709
Mux control													
PC _{out}													
C _{out}													
MD _{out}													
MD _{in}													
R _{out}													
c2 _{out}													
BA _{out}													
MA _{in}													
C _{in}													
A _{in}													
R _{in}													
PC _{in}													
IR _{in}													
NOT													
OR													
Wait													
Read													
AND													
INC4													
Gra													
Grb													
Grc													
End													

Step	RTN for the XOR Instruction	Control Sequence
T0	MA ← PC; PC ← PC + 4;	PC _{out} , MA _{in} , INC4, C _{in}
T1	MD ← M[MA]; PC ← C;	C _{out} , PC _{in} , Read, Wait
T2	IR ← MD;	MD _{out} , IR _{in}
T3	C ← ¬R[rb];	Grb, R _{out} , NOT C _{in}
T4	A ← C;	C _{out} , A _{in}
T5	C ← A ∧ R[rc];	Grc, R _{out} , AND, C _{in}
T6	MD ← C;	C _{out} , MD _{in}
T7	C ← ¬R[rc]	Grc, R _{out} , NOT, C _{in}
T8	A ← C;	C _{out} , A _{in}
T9	C ← A ∧ R[rb];	Grb, R _{out} , AND, C _{in}
T10	A ← C;	C _{out} , A _{in}
T11	C ← A ∨ MD;	MD _{out} , OR, C _{in}
T12	R[ra] ← C;	C _{out} , Gra, R _{in} , End

Microcode Branching Examples

Address	Mux Ctl	BrUn	BrNotZ	BrZ	BrNotN	BrN	Branch Address	Branching Action
200	00	0	0	0	0	0	ddd	None -201 next
201	01	1	0	0	0	0	ddd	To output of PLA
202	10	0	0	1	0	0	ddd	To external address if Z
203	11	0	0	0	0	1	300	To 300 if N (else 204)

24. (30 points) Complete the SRC assembly language program below so that it implements the following C++ statements.

```
int max;
int size = 10;
int nums[10] = {5, 3, -1, 2, 4, 37, -100, 13, -5, 0};
max = nums[0];
for (i = 1; i < size; i++)
    if (nums[i] > max)
        max = nums[i];
```

```
;
```

This program finds the maximum value in an array

```
;
```

```
size:      .org 200
max:       .equ 10
nums:     .dc 100, 3, -1, 2, 4, 37, -100, 13, -5, 0
orig:     .org 1000
```

```
lar  r10, nums          ; pointer to first element of array
la   r1, size           ; holds size of array
ld   r2, 0(r10)         ; max = num[0]
la   r4, 1              ; r4 = 1, index1 into array
```

stop