

EE 2310 Worksheet #9A – Final Exam Review

MIPS Registers

.data
n: .word
p: .word
q: .word
r: .word
s: .word
t: .word
u: .word
v: .word
w: .word
x: .word
y: .word
z: .word
str:
.asciiz
"hello
world\n"

R0 (r0): 0x00000000	R8 (t0): 0x00000000	R16 (s0): 0xff00ff00	R24 (t8): 0x00000000
R1 (at): 0x10010000	R9 (t1): 0x0000ffff	R17 (s1): 0x00000000	R25 (t9): 0x00000000
R2 (v0): 0x00000000	R10 (t2): 0x00000000	R18 (s2): 0x00000023	R26 (k0): 0x00000000
R3 (v1): 0x0000000c	R11 (t3): 0x10010020	R19 (s3): 0x10010000	R27 (k1): 0x00000000
R4 (a0): 0x00000023	R12 (t4): 0x10010030	R20 (s4): 0x00400020	R28 (gp): 0x10008000
R5 (a1): 0x10010010	R13 (t5): 0x10010020	R21 (s5): 0x00000000	R29 (sp): 0x7ffffeff0
R6 (a2): 0x0000000a	R14 (t6): 0x80000010	R22 (s6): 0x800c1001	R30 (s8): 0x00000000
R7 (a3): 0x00000004	R15 (t7): 0xffff0000	R23 (s7): 0x00000010	R31 (ra): 0x00400060

Data

[0x10000000]...[0x1000fffc] 0x00000000

[0x10010000]	0x5350494d	0x20697320	0x61207573	0x6566756c
[0x10010010]	0x2073696d	0x756c6174	0x6f722066	0x6f72206c
[0x10010020]	0x6561726e	0x5350494d	0x4d495053	0x20617373
[0x10010030]	0x68656c6c	0x6f20776f	0x726c640a	0x00000000
[0x10010040]	0x726f6772	0x00000000	0x00000000	0x6e642063
[0x10010050]	0xf7f7f7f7	0xf7f7f7f7	0xf7f7f7f7	0xf7f7f7f7

[0x10010060]...[0x10020000] 0x00000000

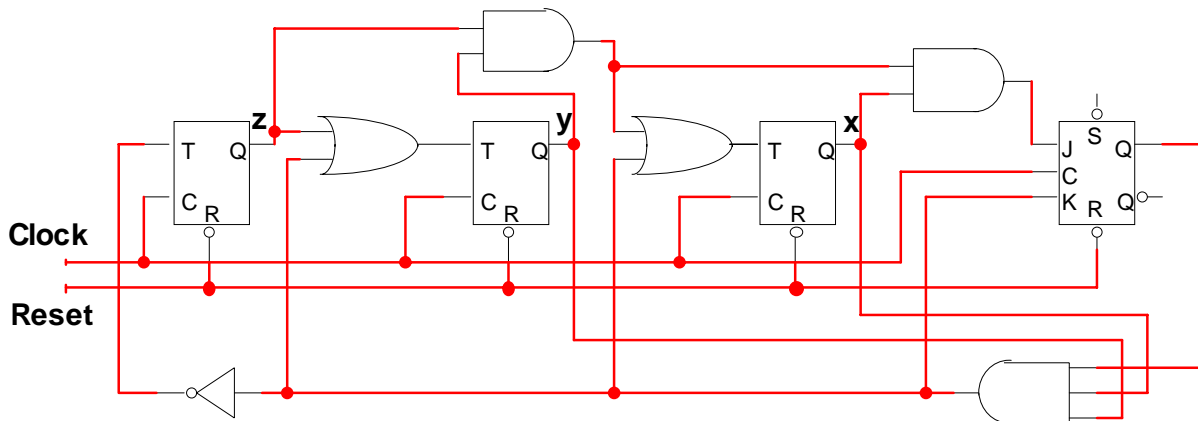
Use the data declaration, SPIM memory dump, and MIPS register readouts above as directed in the problems starting on the following page. Note: If a register or memory location is changed in any problem, the change DOES NOT CARRY OVER to another problem!

Logic Review

4. Draw the simplified circuit for the function
 $f = \Sigma m(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 12, 13) =$
 $f = \Sigma m(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, c, d).$ Use
the Karnaugh map to simplify the function.

		yz			
		00	01	11	10
wx	00				
	01				
	11				
	10				

5. Explain in words how the counter below counts. Hint: The JK FF helps it to achieve its unusual count cycle, and is NOT part of the counter. How does it work? (Note that all flip-flops start off in the “reset” state.)



6. A 4-bit synchronous (parallel) counter that counts modulo-twelve is very useful (there are several in the digital clocks and VCR timers in your home!). Design a modulo-12 4-bit counter, assuming the following:

- The counter is free-running, after a power-up reset.
- The only circuit inputs are a 50/50 clock and the power up reset.
- Use the T ff's below for the four counter elements (they are master-slave type).
- You have available any common logic circuits you need to make the counter.
- Hint: Think how you might use a decoder to help with the reset after count eleven.
- Show the timing below on the diagram.

