Zak Rowland
10/17/2019
CST337
Lab 3 answers

1. (a)(b)(c)(d)

	Status						IntCtl	Cause	INTCON	Ebase	OFF009	OFF014
	BEV	IPL	UM	ERL	EXL	IE	VS	IV	MVEC			
After Power On	1	0	1	1	1	1	0	1	0	0x80000000	0x0	0x0
At beginning of main	0	0	0	0	0	0	0x1	1	0	0x9D000000	0x200	0x200
No defined reset value		U	U	U	U	U		U				

(e) The data sheet explains that the ERL bit is "Set by the processor when a Reset, Soft Reset, NMI or Cache Error exception are taken." Upon reset, the processor decides that the ERL bit should be set to 1.
(f) The processor core is in single vector mode after power on. Single vector mode is selected if Ebase != 0, IV = 1, MVEC = 0, IE = 1, and BEV = 1; these conditions are met after power on.
(g) User mode requires UM = 1, EXL = 0, and ERL = 0, so the processor is in Kernel mode after power on.
(h) The processor core seems to be in none of the interrupt modes because IE = 0 and BEV = 0.
(i) User mode requires UM = 1, EXL = 0, and ERL = 0, so the processor is in Kernel mode after crt.o runs.
(j) The processor priority at the beginning of main is 0. It is important to know the value of the processor priority level because the value can be changed to prevent or allow certain interrupts.
(k) In the Status register, IE needs to be set to 1 and BEV needs to be set to 0.
(l) In the INTCON register, MVEC needs to be set to 1. In the Status register, IE needs to be set to 1 and BEV needs to be set to 0.
(m) crt.o

3.

RUN	1	2	3	4	5	6	7	8	9	10
TMR2	83	28	28	28	28	28	28	28	28	28

The subsequent runs have a lower latency because the routine is likely cached in the CPU. If there were more interrupts or more extensive code, there wouldn't be enough space to store it all in cache so instructions and data would need to be fetched.

Zak Rowland
10/17/2019
CST337
Lab 3 answers

4.

- OFF009 = 0x200
- Computed address = 0x9D000200
- The first instruction is rdpgpr sp, sp

5.

- Address = 0x9D000290
- Label = _DefaultInterrupt
- All other offsets contain 0x290
- If _DefaultInterrupt is executed it will cause a debug breakpoint and reset the device. It's useful to know because the device reset might cause unexpected behavior; the function of _DefaultInterrupt can also be changed.

6. The value of TMR2 after the first run is 17. The latency to this point represents how long it takes to get to the first instruction of the prologue whereas part 3 would go through the prologue first. The value of TMR2 after the second run is 13. The latency is lower because the address is cached, similar to part 3.

7. The opening curly brace will break at the first instruction of the epilogue, and the closing curly brace will break at the first instruction of the epilogue.

8. The following general purpose registers are saved and restored: FP, V1, and V0.

9. Break at beginning of epilogue: Run 1 TMR2 = 124, Run 2 TMR2 = 35. Break at ERET: Run 1 TMR2 = 169, Run 2 TMR2 = 49. Epilogue execution time for the first run is 45 counts, and the execution time for the second run is 14 counts.

10.

RUN	1	2	3	4	5	6	7	8	9	10
TMR2	71	23	23	23	23	23	23	28	28	28

(Step 3) The time from step 3 went down from the original because priority 7 can't be interrupted, this means the EPC save and restore can be skipped as well as saving the general purpose registers.
(Step 6) The value of TMR2 after the first run is 17. The value of TMR2 after the second run is 13. The latency is similar but it might increase by a cycle or two because the processor has to switch to using shadow set registers.
(Step 8) The only general purpose register saved and restored is FP.

11. The value in OFF009 is 0x200 and the value in OFF014 is 0x290. The timer counts are in sync.

Zak Rowland
10/17/2019
CST337
Lab 3 answers

12.
Expected:

RUN	1	2	3	4	5	6	7	8
Handle	T2	T3	T2	T3	T2	T3	T2	T3

Actual:

RUN	1	2	3	4	5	6	7	8
Handle	T2	T2	T3	T2	T3	T2	T2	T3

Timer2 is interrupting and nesting in Timer3's routine. The Timer3 routine is still in the prologue when Timer2's interrupt is triggered again.

15.

RUN	1	2	3	4	5	6	7	8	9	10
TMR2	67	75	28	51	28	28	28	28	28	28

(a) Timer2 and Timer3 access some of the same data in the prologue which allows the data cache to be utilized since Timer3 had an artificial interrupt.
(b) Since Timer3 now has higher priority, Timer2 has to now wait for Timer3's routine to finish.
(c) Timer3 has 24 additional cycles to finish during run 4 compared to run 2.

16. Set breakpoints on the opening braces and first instruction of each routine. This allows us to see the Timer3 prologue being hit but being interrupted by Timer2, specifically on the 3rd run.

17. (a) RIPL can handle 256 vectors, and in single vector mode natural priority is used.
(b) Local variables are instanced within the scope of its function, so separate copies of the variables are stored.
(c) The interrupt flags assigned to var won't be up to date so the wrong interrupt will be serviced first.