

This print-out should have 11 questions. Multiple-choice questions may continue on the next column or page – find all choices before making your selection. The due time is Central time.

EE345L Valvano Homework 3.

---

**001** (part 1 of 1) 10 points

What property of a debugging activity best describes the measure to which the debugging code itself affects the software it is trying to evaluate?

1. Hierarchical.
2. Static efficiency.
3. Modularity.
4. Nonintrusiveness.
5. None of these choices is correct.
6. Stabilization.

---

**002** (part 1 of 1) 10 points

Consider the situation where you are trying to measure the number of characters processed per second. Assume the correct rate is 1000 characters per second, but it takes so much time to run the debugging code that the processing rate actually drops to 500 characters per second.

What term describes this debugging mistake?

1. Overflow.
  2. Unstable.
  3. Underflow.
  4. Dropout.
  5. None of these choices is correct.
  6. Intrusive.
- 

**003** (part 1 of 1) 10 points

What is a **logic analyzer**?

1. A multiple channel digital storage scope with many ways to trigger.
2. None of these choices is correct.
3. A device that emulates the microcomputer.
4. Software application to analyze digital logic functions.
5. An analog oscilloscope.
6. A digital voltmeter.

---

**004** (part 1 of 1) 10 points

What is an **emulator**?

1. A debugging technique used to implement breakpoints.
2. A software application used to provide debugging functions.
3. None of these choices is correct.
4. Software application that simulates the hardware and software functions of the microcomputer.
5. A multiple channel digital storage scope with many ways to trigger.
6. A hardware debugging tool that recreates the input/output signals of the processor chip.

---

**005** (part 1 of 1) 10 points

What is the **background debug module (BDM)**?

1. A **BDM** is a software application, which runs in the background, that simulates the hardware and software functions of the microcomputer.

**2.** A **BDM** is software code added to the program to visualize time-dependent execution, which runs in the background.

**3.** A **BDM** is a hardware built into the processor itself and an external POD. Together they provide debugging features such as observing memory while the program is running.

**4.** None of these choices is correct.

**5.** A **BDM** is a multiple channel digital storage scope with many ways to trigger.

**6.** A **BDM** is a hardware debugging tool that recreates the input/output signals of the processor chip.

---

**006** (part 1 of 1) 10 points

To verify the proper functionality of a subroutine, we need to record its input and output parameters. In particular we write a special main program that provides a known and repeatable sequence of inputs to the subroutine under test. In this way, each time the test subroutine is changed, we can be sure the change in output values is caused by the software modification and not due to a change in input values.

What is this debugging procedure called?

- 1.** Monitor.
- 2.** Stabilization.
- 3.** None of these choices is correct.
- 4.** Nonintrusive debugging.
- 5.** Performance debugging.
- 6.** Emulation.

---

**007** (part 1 of 1) 10 points

What is the major difference between **performance debugging** and **functional de-**

**bugging**?

**1.** Performance debugging evaluates the interactive between functions, and functional debugging measures the accuracy of the system.

**2.** Performance debugging studies the time behavior, and functional debugging tests if the proper output values are produced.

**3.** Functional debugging evaluates the interactive between functions, and performance debugging measures the accuracy of the system.

**4.** None of these choices is correct.

**5.** Functional debugging studies the time behavior, and performance debugging tests if the proper output values are produced.

**6.** They are two names that mean the same thing.

---

**008** (part 1 of 1) 10 points

Which method can be used to measure the **execution speed** of a function?

**1.** First determine how many times each assembly language instruction in the function is typically executed. E.g., instructions that are not usually executed get a count of zero. Then, multiply these counts by the corresponding number of cycles each instruction takes to execute. Add them up to get the typical cycle count for each run through the function. Lastly, multiply by the time per cycle for this microcomputer.

**2.** Each of the other three methods can be used under certain conditions.

**3.** Assume **TCNT** is actively counting. Execute this program, and observe the value in **Time**.

```
unsigned short Begin;
unsigned short Time;
void test(void){
```

```

Begin = TCNT;
funct();
Time = TCNT-Begin;
}

```

4. Attach an unused output pin, PT0, to an oscilloscope. Run this program, and measure the pulse width of PT0 using the scope.

```

void main(void){
    DDRT |= 0x01;
    while(1){
        PORTT |= 0x01;
        funct();
        PORTT &= 0xFE;
    }
}

```

---

**009** (part 1 of 1) 10 points

Which answer best describes **profiling**?

1. None of these choices is correct.
2. **Profiling** involves the dynamic measurements of how fast a subroutine executes.
3. **Profiling** is the creation of a profile of the computing resources required of the software.
4. **Profiling** is the analysis of the data flow graph creating a profile of the data values as they are processed.
5. **Profiling** is the collection of information that involves which functions were executed, when they were executed, and what data were they processing.
6. **Profiling** is the analysis of the call graph creating a profile of which functions call which other functions.

---

**010** (part 1 of 2) 10 points

Consider the following function as implemented by the Metrowerks CW12 compiler.

```

short add1(short input){
    return(input+1);
}

```

At the time the branch to subroutine instruction, **bsr**, is being executed, where is input parameter located?

1. global RAM
2. EEPROM
3. stack RAM
4. Register B
5. Register X
6. Register D
7. Register A

---

**011** (part 2 of 2) 10 points

At the time the return from subroutine instruction, **rts**, is being executed, where is return parameter located?

1. EEPROM
2. global RAM
3. Register A
4. Register D
5. Register X
6. stack RAM
7. Register B