Tyler Punch

ECE471 Homework 1

1a) This is an embedded system for several reasons, the obvious one being that this system is ‘embedded’ within an electric toothbrush. This is also clearly a fixed-purpose system meant solely to turn the toothbrush motor on and off with the pressure sensor. It has minimal hardware and takes up very little space since it can fit inside a toothbrush. Also, turning a motor on and off at the push of a button does not require a lot of code. This device is also resource constrained. An 8-bit system is a small one compared to the 64-bit systems that are used in general purpose technology like the laptop I’m writing this on. To accompany that this system does not have a lot of memory either at only 512 bytes of RAM. This is an embedded system because all the characteristics mentioned comply with those of an embedded system.

1b) This device is not an embedded system; it is a general-purpose computer. An embedded system should be resource constrained with a small CPU and less memory. This device has a six-core 64-bit CPU which is by no means a small CPU. That is more cores than I have in my laptop. It also has 3 GB of RAM and 3 levels of cache memory which would not be necessary for an embedded system. An embedded system should also be fixed purpose consuming less power, with less hardware and relatively low cost. Since this CPU is very powerful accompanied by a powerful graphics card it certainly consumes a lot of power. All the hardware mentioned above shows that also has too much hardware to be an embedded system. Finally, even though the iPhone 8 is 2 or 3 models behind the current model, is still runs for about $600 which is not low-cost. Therefore, this is not an embedded system because it does not comply with the characteristics that define an embedded system.

1c) This is an embedded system because it is a fixed-purpose, resource-constrained and embedded inside something. This device is fixed purpose because it does not have much hardware or code, it does not take up a lot of space and is low power. The fixed purpose is to monitor and control the temperature of the room this device resides in. The keypad and is simply to set the temperature and maybe a few other instructions such as putting the heat on a timer. The LCD display would simply display the current temperature or the desired temperature and possibly a few select words such as heat or cool. The sensor and relay only serve a single purpose and that is to turn the HVAC system on or off depending on the temperature of the

sensor. Very simple and certainly fixed purpose. It is also resource constrained with only a small 8-bit PIC processor that only runs at 10MHz. Since the operation of this system is so simple it would not need a lot of memory to hold the code it requires for operation nor does it have to keep track of much data save for the few settings it has. Lastly, an obvious reason this is an embedded system is since it is within a wall thermostat, meaning it is literally embedded and small.

2a) This system is a 32-bit system because it's registers and ALU are 32-bit. This means that all the data handled in the system is 32-bit data so that means that everything this system does is in 32-bit.

2b) This system is an 8-bit system because the registers, data bus and ALU are all 8-bit. Even though the instruction pointer and address bus are 16-bit, this system processes data in 8-bits so it makes the most sense to call this an 8-bit system.

3a) Using an ASIC may be better than using a micro-controller because ASIC's are faster than microcontroller at performing a specific operation. If I am designing an embedded system that is hard real-time constrained, then I would want something that can perform fast enough to ensure the safety of whoever needs this system. For example, an emergency stop on a machine would need to execute that operation as fast possible to ensure the expensive machine isn't damaged and people don't get hurt.

3b) Using a microcontroller may be better than using an ASIC if whatever embedded system I'm designing must be mass produced. This is because if a mistake is made in the design of an ASIC then all the devices would need to be recalled which would be next to impossible. This would also mean that the company I'm working for would lose a lot of money and could significantly impact consumers as well depending on how vital this embedded system is. With a microcontroller I could simply patch the mistake in the software and send it out quickly meaning no real loss financially and no waste of time and resources rebuilding the system.