**MEJAH JOHN ONYANGO**

**BTIT/041J/2013**

**EMBEDDED SYSTEM ASSIGNMENT 1**

**Challenges faced when designing an embedded system (10mrks)**

**Harsh environment**

Many embedded systems do not operate in a controlled environment. Excessive heat is often a problem, especially in applications involving combustion (*e.g.,* many transportation applications). Additional problems can be caused for embedded computing by a need for protection from vibration, shock, lightning, power supply fluctuations, water, corrosion, fire, and general physical abuse.

## Cost sensitivity

Even though embedded computers have stringent requirements, cost is almost always an issue . Although designers of systems may talk about the importance of cost with equal urgency, their sensitivity to cost changes can vary dramatically. A reason for this may be that the effect of computer costs on profitability is more a function of the proportion of cost changes compared to the total system cost, rather than compared to the digital electronics cost alone. However, with in the Small system decisions increasing costs by even a few cents attract management attention due to the huge multiplier of production quantity combined with the higher percentage of total system cost it represents.

## End-product utility

The utility of the end product is the goal when designing an embedded system, not the capability of the embedded computer itself. Embedded products are typically sold on the basis of capabilities, features, and system cost rather than which CPU is used in them or cost/performance of that CPU.

## System safety & reliability

A bigger and more difficult issue at the system level is software safety and reliability. While software doesn't normally "break" in the sense of hardware, it may be so complex that a set of unexpected circumstances can cause software failures leading to unsafe situations. This is a difficult problem that will take many years to address, and may not be properly appreciated by non-computer engineers and managers involved in system design decisions (discusses the role of computers in system safety).

## Power management

A less pervasive system-level issue, but one that is still common, is a need for power management to either minimize heat production or conserve battery power. While the push to laptop computing has produced "low-power" variants of popular CPUs, significantly lower power is needed in order to run from inexpensive batteries for 30 days in some applications, and up to 5 years in others.

## Component acquisition

Because an embedded system may be more application-driven than a typical technology-driven desktop computer design, there may be more leeway in component selection. Thus, component acquisition costs can be taken into account when optimizing system life-cycle cost. For example, the cost of a component generally decreases with quantity, so design decisions for multiple designs should be coordinated to share common components to the benefit of all.

## System certification

Embedded computers can affect the safety as well as the performance the system. Therefore, rigorous qualification procedures are necessary in some systems after *any* design change in order to assess and reduce the risk of malfunction or unanticipated system failure. This additional cost can negate any savings that might have otherwise been realized by a design improvement in the embedded computer or its software. This point in particular hinders use of new technology by resynthesizing hardware components -- the redesigned components cannot be used without incurring the cost of system recertification.

One strategy to minimize the cost of system recertification is to delay all design changes until major system upgrades occur. As distributed embedded systems come into more widespread use, another likely strategy is to partition the system in such a way as to minimize the number of subsystems that need to be recertified when changes occur. This is a partitioning problem affected by potential design changes, technology insertion strategies, and regulatory requirements.

## Logistics and repair

Whenever an embedded computer design is created or changed, it affects the downstream maintenance of the product. A failure of the computer can cause the entire system to be unusable until the computer is repaired. In many cases embedded systems must be repairable in a few minutes to a few hours, which implies that spare components and maintenance personnel must be located close to the system. A fast repair time may also imply that extensive diagnosis and data collection capabilities must be built into the system, which may be at odds with keeping production costs low.

Because of the long system lifetimes of many embedded systems, proliferation of design variations can cause significant logistics expenses. For example, if a component design is changed it can force changes in spare component inventory, maintenance test equipment, maintenance procedures, and maintenance training. Furthermore, each design change should be tested for compatibility with various system configurations, and accommodated by the configuration management database.