Lect 5
Chapter # 2

Measurement Techniques

How does one measure a system or component performance? This is the main problem facing the computer systems performance analyst. To determine how to measure, when to measure, or what to measure, the analyst must first know all of the events of interest in the system and the relationship these events have with each other.These events form a hierarchy of relationships, where the finer, granular events are used to construct the coarse-grained events in the system.-book 2 pg 112

• How does one measure a system or component performance?
• The analyst must first know all of the events of interest in the system and other relationship these events have with each other.
• These events form a hierarchy of relationships, where other finer, granular events are used to construct the coarse-grained events in the system.-vedio

The state of system:

It is defined by values contained in various storage elements, i.e. memory locations, registers or flip flops.

- Depending on measurement objectives, some of these may be used to define relevant states and others to provide further information about happenings in those states.
- We call the former primary variables and others auxiliary variables.
- For e.g., the relevant states may be only the busy/idle condition of the paging disk, but we may be interested in knowing the no. of free blocks whenever a page I/O is initiated.-vedio

So in this particular case the busy or ideal condition is actually representing the primary condition & the number of free blocks is actually representing the auxilary variable.

Event:

It refers to a change in relevant state variable.

- The events could also be classified as primary or auxiliary depending on what type of state variables are involved.-vedio

Three Primary Types of Measures-book 2 pg 113

• **Type A** looks to count the number of times a given state is visited during a given time period.

• Some of the examples are:

- the number of times a data structure is referenced,
- relative frequency of executing a given instruction,
- No. of times I/O done from cylinder 0 of some disk etc.
- In the last example, the relevant state is defined by the situation where I/O is in progress on cylinder 0 and the primary event is the initiation of such an I/O operation.-vedio

• **Type B** measures all these state variables.

- e.g. to extract all of the values for all internal registers and devices at the beginning of an instruction execution cycle.
- Another example is the number of processes in the ready list whenever an I/O operation is initiated.-vedio

• **Type C** measures the fraction of time the system is within a state.

- e.g. the fraction of time the system is executing load instructions versus all other kinds of instructions during the measured period of time.

- Another example, what fraction of the time the disk head stays on cylinder 0.-vedio

missing...

Issues in Measurement

Two issues in measurement:
• How do we recognize conditions for measurement?
• How do we actually measure the quantity of interest?-vedio

Conditions for Measurement

The condition for measurement can be recognized in two ways:
1) As being in a relevant state:
• A natural way to do this is to sample the system and check if the primary state variables have the desired values.
• For example, to check if the control is inside a given procedure, we sample the program counter and see if contains an address that belongs to that procedure.
• This leads to sample monitoring.
2) As an event that brings the system to a relevant state:
• To check if control is inside a given procedure, we explicitly look for events of entry to and exit from the procedure.
• This leads to trace monitoring.-vedio

Remarks

• It is easy to see that we can do measurements of all three types using trace monitoring.

- Type A measurements cannot be done using sampling.

Recall type A measurements, it actually count the number of times a given state is visited during a given time period. Since sampling would fail to count all the instances when the system was in the desired state even the relative frequencies can not be estimated properly by sampling because if two consecutive samples shown the system to be in same state if we can not tell if the system has been in that state or is visiting in the next time. However if the duration of the desired state is shorter than the inter sampling gap the relative frequencies can be estimated reliably.

- Similar comments apply to type B measurements.
- Type C measurements are possible by sampling since the computed relative frequency really gives an estimate of the fraction of time spent in the desired state.

Multiplying this by the measurement duration we can also get an estimate of total time spent in the desired state.

Sampling only gives an estimate and not the exact answer.

- It may appear, from the discussion, that trace monitoring is always preferable to sampling, but this is not true.-vedio

Classification of Instrumentation

- Hardware Monitoring,
- Software Monitoring,
- Hybrid monitoring

The decision about which of these techniques to use is dependent on many factors, such as accessibility, event frequency, monitor artifact, overhead of monitoring, and the flexibility of the technique used.-book 2 pg 113

“A monitor is a tool used to observe the activities on a system. In general, monitors observe the performance of systems, collect performance statistics, analyze data, and display results”-vedio

Monitors are used not only by performance analysts but also by programmers and systems managers.
Some of the reasons to monitor a system are as follows:
• To find frequently used segments of software and optimize their performance.
• To measure resource utilizations and to find performance bottlenecks.
• To tune the system. The system parameters can be adjusted to improve the performance.
• To characterize the workload. The results may be used for capacity planning and for creating test workloads.
• To find model parameters, to validate models, and to develop inputs for models.-vedio

Hardware Monitoring

• Employs additional monitoring hardware that is interfaced with the system under measurement in a nonintrusive way.
intrusive means that something is coming that is not being invited or something is not welcomed. It means that something is distracting or distrubing the original operation or any original procedure. So nonintrusive means that if we are actually interfacing an additional hardware with our system i.e. to be measured. If that particular additional hardware wont disturb or distract the original operation i.e. in progress or i.e. in process of that particular system.
• For example:
• we might attach a logic analyzer to measure the signals within the system or

- we may insert a specially designed hardware card to extract some signals from a system.
- We can only measure what is exposed and available to be attached to for monitoring.

- Hardware monitors consist of:
  - a set of probes or sensors,
  - a logic-sensing device,
  - a set of counters, and
  - a display or recording unit.-vedio

- The probes monitor the state of the chosen system points.
- Typically, probes can be programmed to trigger on a specific event, thereby providing the ability to trace specific occurrences within a system.
- The logic-sensing subsystem is used to interpret the raw input data being probed into meaningful information items.
- The counters are used to set sampling rates on other activities requiring timed intervals.
- The last component records and displays the information as it is sensed and reduced.
- The ability to perform effective operational analysis is directly dependent on the hardware and software monitors' ability to extract information.

The hardware or software monitors that are being placed with the actual system to be measured they must be able to extract all the required information. So that the system manager or the system analyst can effectively perform there analyse's operation.

- Another form of hardware monitoring uses integral test hardware, designed into the system being monitored during systems design.

integral test hardware means that this is some testing hardware or monitoring hardware tool i.e. fundamental & essential & is already designed into the system being monitored. So we dont need some extra hardware monitoring setup for monitoring our system. Its already present there in the system & it was designed during the system's designed process.

- Many VLSI devices are designed so that:
- all data items of interest can be tested in the device itself, or,
- at a minimum, the test data points are brought outside of the chip
- so additional devices can be used to gather this information and compute the health of the device.
- In all of these cases it is imperative that the hardware monitoring be designed as an integral component of the system, so it will not interfere with the operational system.-vedio

Requirements for Hardware Monitoring

It is not desirable for the monitoring equipment to interfere with the system being monitored. If this is the case, the results from the monitoring are suspect and may lead to erroneous conclusions..-book 2 pg 129
The determination of sampling sites and the frequency of measurements must be designed ahead of time, not after the monitor has been put in place. The monitoring method has to be set up ahead of time also. That is, we must determine if the monitor is to act synchronously or asynchronously with the measured system. We must determine and define all aspects of the monitor's existence in the measured system.-book 2 pg 129

- It is not desirable for the monitoring equipment to interfere with the system being monitored.
- The determination of sampling sites and the frequency of measurements must be designed ahead of time, not after the monitor has been put in place.
- The monitoring method has to be set up ahead of time also.

• We must determine and define all aspects of the monitor's existence in the measured system.-vedio

Detailed Mechanism - Hardware Monitoring

• The conditions for measurement are indicated by a logic signal S, synthesized using more primitive logic signals available from the machine backplane.

Synthesize means that we want to produce something electronically so here i want to synthesize a logic signal S using some primitive logic signals.

what are the pritimive logic?

primitive logic includes the buffer, and gate, or gate, nand, nor, xor and so on. These are called primitive logic or primitive logic gates or simply the gates.

backplane is a computer bus .

• S may be synthesized as both single-bit and multi-bit signals.

• In addition to normal boolean functions, the synthesizer should provide functions like comparison of multi-bit signals, AND of all bits, OR of all bits, etc.

Assumptions:

• a 0 to 1 transition in S takes to a relevant state.

• a 1 to 0 transition takes out of that state.

• e.g., suppose the condition of interest is the simultaneous operation of two devices, say D1 and D2.

There are two devices D1 & D2 & we want to know that when both the devices are in operational state. So let suppose that Di is representing the device & the busy condition for that particular device is indicated by

• Busy condition for Di indicated by boolean flag Fi in the backplane.

so this signal can easily be calculated or evaluated by means of this

boolean function
Then: S = F1^F2
when both the devices will be operational the values of F1 & F2 will be 1, Hence the resulting signal S will also be 1.
If any one of the devices at any pointing time is not operational then the value of S will be 0.
• If there is more than one such state, a multi-bit signal, S',   that can be synthesized using auxiliary state variables.-vedio

For example: if we wish to determine the
Relative Frequency of executing each instruction:

• a 0 to 1 transition in S may indicate a new instruction has been fetched into the instruction register, and
• S' is the set of bits that represent the opcode.

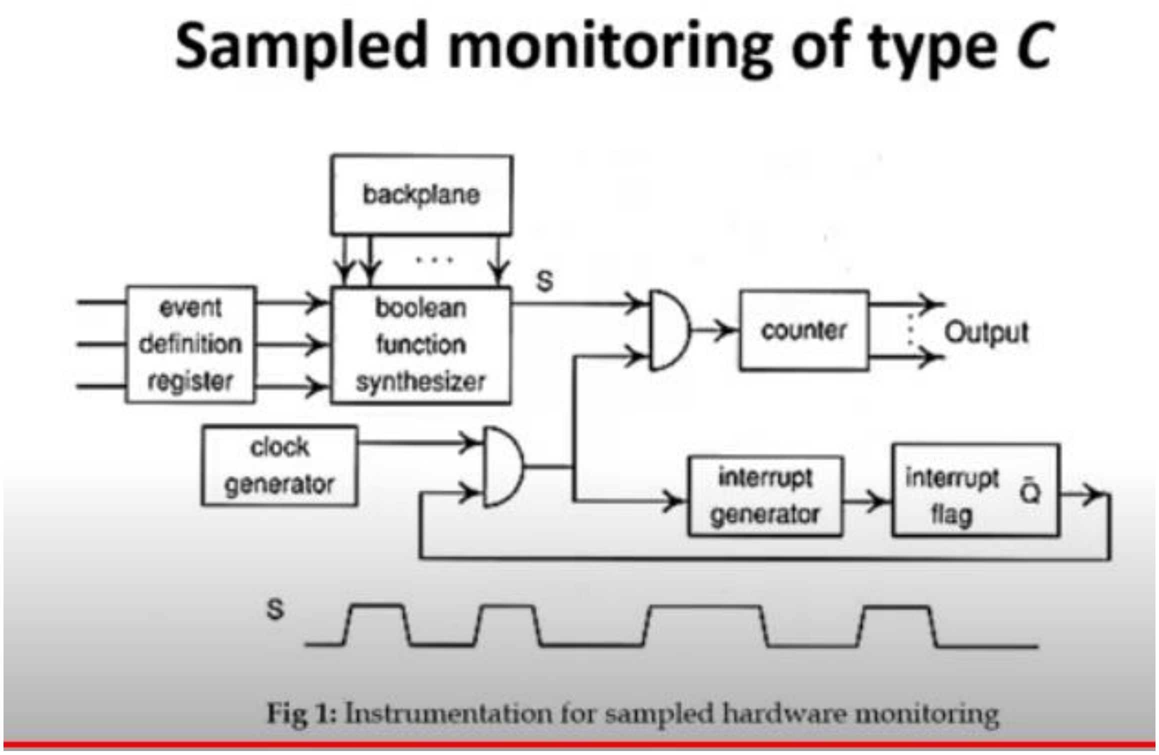
Type A
• increment a counter whenever S makes a 0 to 1 transition.
• The counter is chosen from an array indexed by S'.
• The signal S' may be valid only when S = 1 as in the instruction frequency example above.
when the 0 to 1 transition is completed & the value of S is 1, it means that the instruction has been fetched into the instruction register. So the S' will only be valid when the transition is being completed.
Type B
• leading edge of S used to transfer auxiliary state information from the backplane to a memory module part of the monitor.
As, before the execution of the instruction or when the execution has been started or when the new instruction has been fetched into the instruction register the initial values of the registers will be updated. So those values are already present in the internal registers or the

device registers & that particular information can be collected from the computer bus through the through the memory part of the monitor. That particular information is called auxiliary state information.

Type C

- In discussing type C measurement, assume that S is already synthesized appropriately.
- The measurement can be done using either the trace or the sampling technique.
- In the trace technique, we can accumulate the duration for which S = 1 by keeping track of the times when S makes 0 to 1 and 1 to 0 transitions but that’s nearly impossible considering rapid rate at which hardware signals change.
- Because of this, type C measurements in hardware are almost invariably done by sampling.
- The idea is to use S to gate clock pulses, which are then counted by a hardware counter.-vedio

Sampled monitoring of type C



Sampled monitoring of type *C*

Fig 1: Instrumentation for sampled hardware monitoring

In this figure we are actually representing sampled hardware monitoring.
At the begining of the measurement interval the value of counter is set to 0.
Lets suppose there's a counter variable C & initially its value is going to be 0.
C=0
There's a 'event definition register' (EDR) so the boolean function that defines the event in terms of the raw input signal itself selected by putting the appropriate code in this EDR.
There's an 'interrupt generator' circuit that detects the end of the measurement period & sets the 'interrupt flag'.
So setting the interrupt flag freezes the counter.
As you can see from this circuitry, the counter value at this point can be read by the interrupt handler.
So if the purpose of the experiment is to measure the section of time it

is desirable to choose the measurement interval as 2^N clock pulses where N could be the number of bits in the counter

& it is also preferable to let the measured system interact with the monitor at the start & end of the measurement section.

So this requires to facilities:

1) there should be a machine instruction or a system call that starts the measurement.

2) the CPU should be able to recognize the interrupt posted by the measurement circuit & then read the value contained in the counter.

So it means that the hardware monitor would have an entry in the interrupt factor of the machine & the event definition register i.e. EDR & the counter would appear like command & data registers to the CPU.

So with all these facilities in place we can determine the fraction of time of the desired measurement session.