Honeywell

Analyze UCN Node Schedule

L5685
UCN

TotalPlant®

Copyright, Notices, and Trademarks

Revision 03 – March 9, 1998

Honeywell IAC courseware is subject to change without notice.

This module supports **TotalPlant** Solution (TPS) system network.

TPS is the evolution of TDC 3000^{X}.

Honeywell
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85023

1-800-852-3211

Table of Contents

Figures and Tables

Acronyms

AM ..... Application Module
APM ..... Advanced Process Manager
APMM ..... Advanced Process Manager Module
ARM ..... Archive Replay Module
CG ..... Computer Gateway
CL/APM ..... Control Language/Advanced Process Manager
CL/PM ..... Control Language/Process Manager
IOL ..... Input Output Communication Link
IOP ..... Input Output Processor
LCN ..... Local Control Network
LM ..... Logic Manager
NIM ..... Network Interface Module
PM ..... Process Manager
PMM ..... Process Manager Module
PPX ..... Point Processing
PU ..... Processing Unit
UCN ..... Universal Control Network
US ..... Universal Station

References

Publication Title	Publication Number	Binder Title	Binder Number
For R5xx :			
Universal Control Network Guidelines	UN20-500	Installation/UCN	TPS 3041
HPM Implementation Guidelines	HP12-500	Implementation/HPM-1	TPS 3066-1
APM Implementation Guidelines	AP12-500	Implementation/APM-1	TPS 3042-1
PM Implementation Guidelines	PM12-500	Implementation/PM-1	TPS 3040-1
HPM Control Functions and Algorithms	HP09-500	Implementation/HPM	TPS 3066-1
APM Control Functions and Algorithms	AP09-500	Implementation/APM	TPS 3042-1
PM Control Functions and Algorithms	PM09-500	Implementation/PM	TPS 3040-1
For R4xx:			
UCN Guidelines	UN20-400	Installation/UCN	TPS 2041
APM Implementation Guidelines	AP12-400	Implementation/APM	TPS 2042-1
PM Implementation Guidelines	PM12-400	Implementation/PM	TPS 2040-1
APM Control Functions and Algorithms	AP09-400	Implementation/APM	TPS 2042-2
PM Control Functions and Algorithms	PM09-400	Implementation/PM	TPS 2040-2

Introduction

Module Overview

About this module

This course module discusses the schedule display for the Advanced Process Manager and how it can be used to interpret, monitor, and troubleshoot UCN node functions. Concepts discussed herein also apply to the Process Manager unless otherwise noted.

Objectives

The objectives of this course module are to

- Interpret the information displayed in the schedule display such as
  - Slot execution order,
  - Execution subcycles, and
  - Overruns.
- List ways the display is used to monitor performance.

Sample test items

This course module's Criterion Test includes the following items:

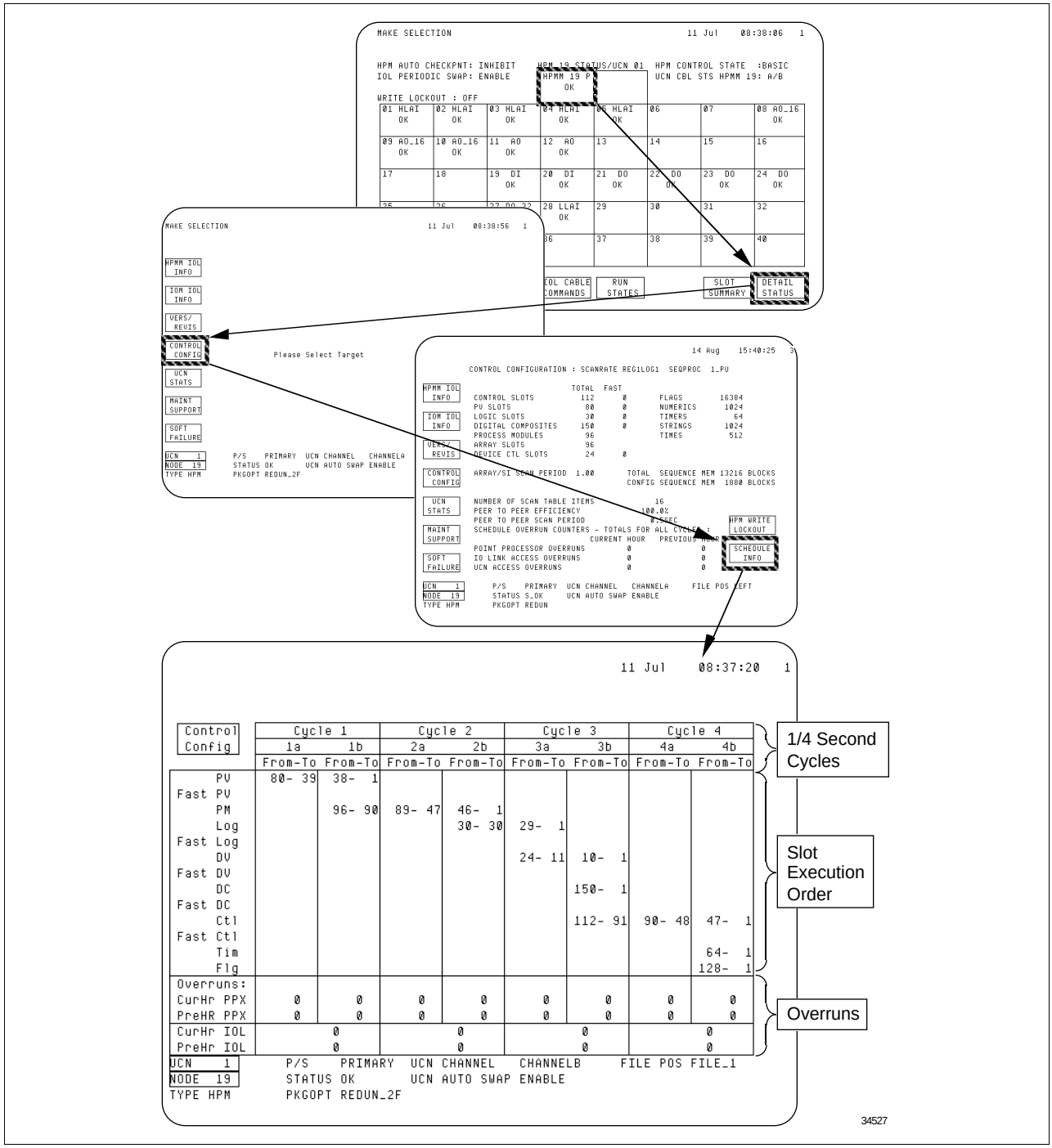
- Describe a scenario where the schedule display would be used to identify a less than optimum control strategy design. You can use an example from the course material, or describe an example you think would not be the best strategy.
- Point to the information displayed in the schedule display such as
  - Slot execution order,
  - Execution subcycles, and
  - Overruns.

Call Up the Schedule Display

Overview

Introduction The navigation path to the schedule display is shown in Figure 1. The display is typically used to identify what cycle a point is executed on.

Figure 1 Schedule Overview

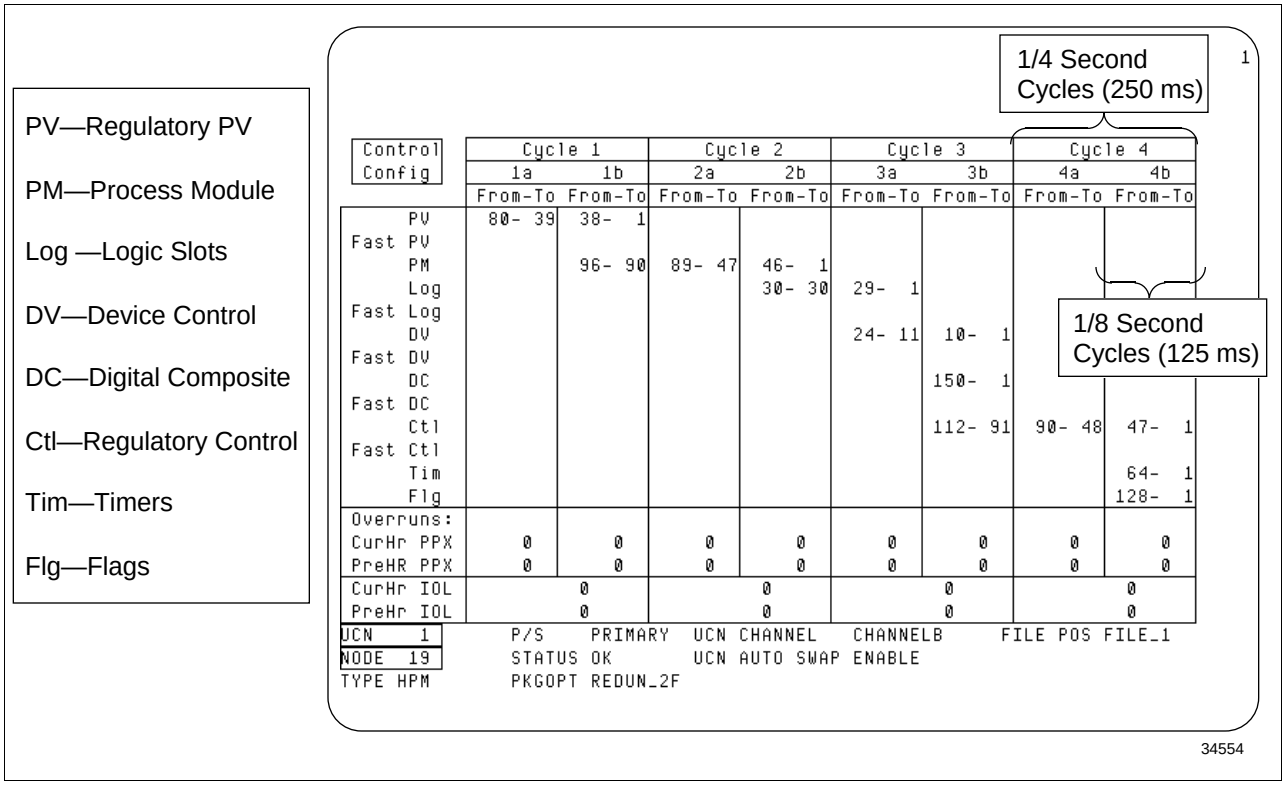


Description of Cycles

Interpret cycles

Within the 1/4 second cycle, there are two subcycles. The best way to look at the point scheduling displays is to think of eight subcycles running in 1/8 of a second. All of the points scheduled in the subcycle run one after the other in descending order, from Regulatory PV through process modules, logic, device control, digital composite, and regulatory control. There are no pauses within a subcycle. All points run consecutively. Refer to Figure 2 for an example display.

Figure 2 Cycles and Subcycles



Rule of thumb

Two good rules of thumb are

- because each APM/PM has 160 PUs (processing units), each subcycle represents approximately 20 PUs, and
- because each HPM has 800 PUs (processing units), each subcycle represents approximately 100 PUs.

Use the Schedule Display

Identify Execution Order

Introduction

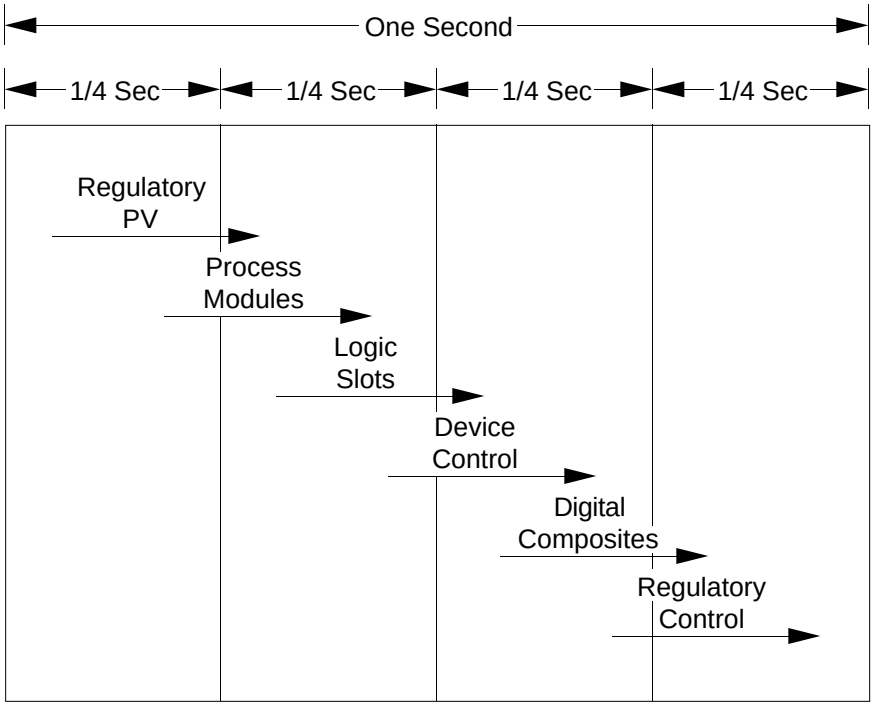
The point (slot) execution order is the most obvious information presented on the schedule display. As you would expect, the point execution is automatically scheduled by the UCN node. The schedule display is also used to do the following:

- Identify point processing or I/O Link overruns on a particular cycle.
- Identify how a point's scheduled execution interacts with another's.
- Identify opportunities for monitoring peer-to-peer communication.

Quick review

Figure 3 is a graphical way of representing the schedule display shown in Figure 2. As Figure 3 illustrates, the points are executed in descending order.

Figure 3 Point Execution Order

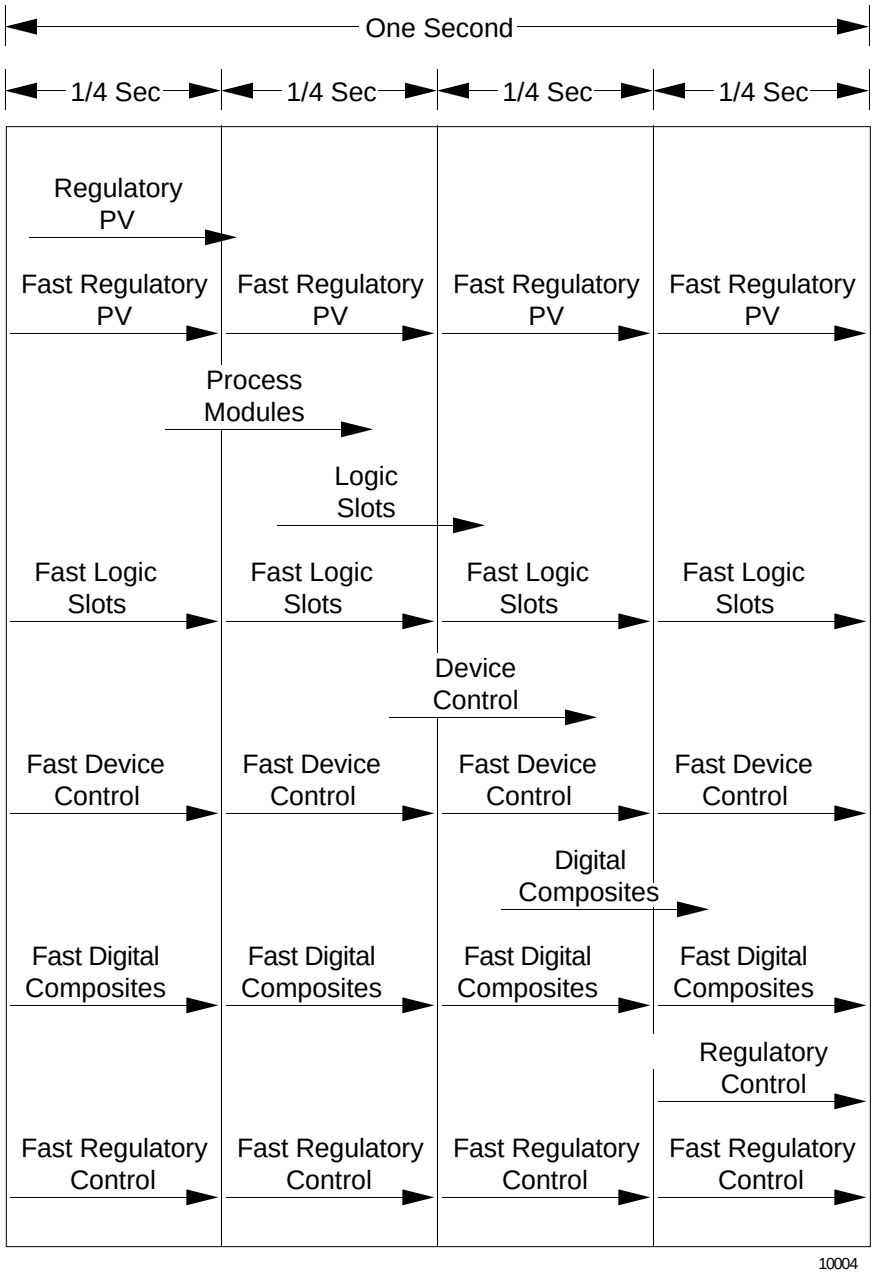


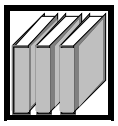
Continued on next page

Fast slots

When fast slots are configured, the point execution order should actually appear as Figure 4.

Figure 4 Schedule with Fast Slots





REFERENCE—Section 3 of the*Control Functions and Algorithms* manual provides a full description of the overrun status information. The information is summarized next for your convenience.

Identify Point Before/After Schedule

Introduction

On some occasions, you may wish to review the execution order of points that are used in a control strategy (such as a cascade loop). For example, point B may be needing data from another point A. If point B is executed after point A, point B will have the most current data; however, if point B is executed before point A, the transfer of data may not be as seamless as desired.

Before/after usually OK

Most of the time before/after scheduling is not a problem. This is because most users build points in ascending slot order. At point execution time, the slots are executed in descending order.

For example, in a cascade arrangement, the secondary control loop is usually built first as a lower slot, then the primary control loop is built as the next higher slot. When the cascade loops are commissioned, the primary loop executes first, then the secondary. That way the secondary loop has the most current data available at point execution time.

Identify Prefetch and Poststore Schedule

Introduction

When control points execute, they often require data from an AI or AO IOP; consequently, a prefetch and poststore cycle occurs on the following:

APM/PM

- on AI and AO points

HPM

- on AO points.

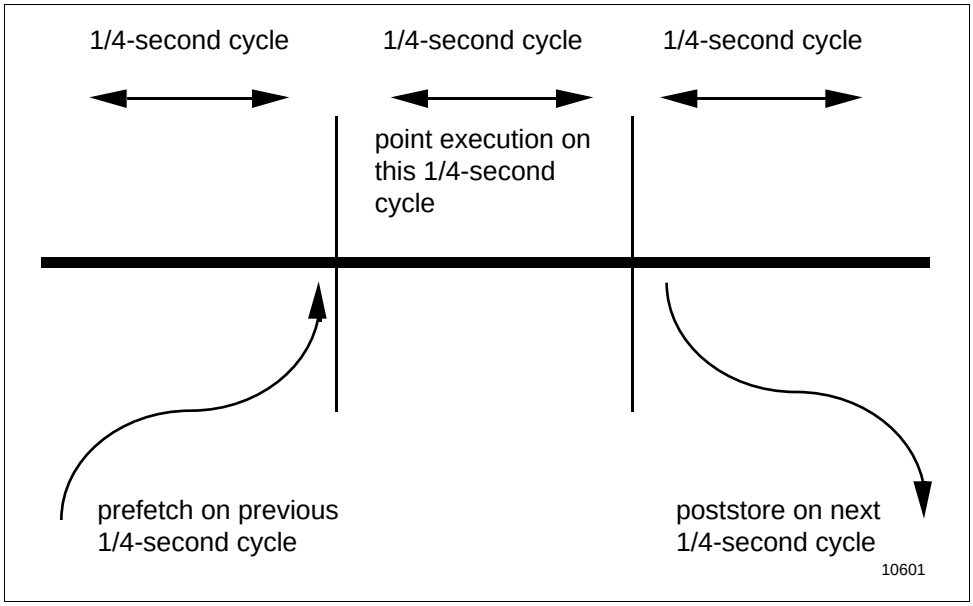
Prefetch/poststore

When a control point using IOP data executes, the following occurs:

- prefetch occurs on the previous 1/4 cycle (APM/PM AI/AO and HPM AO),
- point execution occurs, then
- poststore occurs on the next 1/4 cycle (APM/PM AI/AO and HPM AO).

Figure 5 illustrates the cycles.

Figure 5 Prefetch and Poststore Cycles



Digitals scanned independently

For the APM and PM, the prefetch and poststore cycles apply to AI and AO prefetch and poststore cycles.

For the HPM, prefetch and poststore cycles apply to AO only. The HPM scans AIs every 1/4 second (except LLMUX points scanned every 1/2 second and SI points scanned at the configured rate).

Digital inputs and digital outputs are scanned every 1/4 second.

Identify Overruns

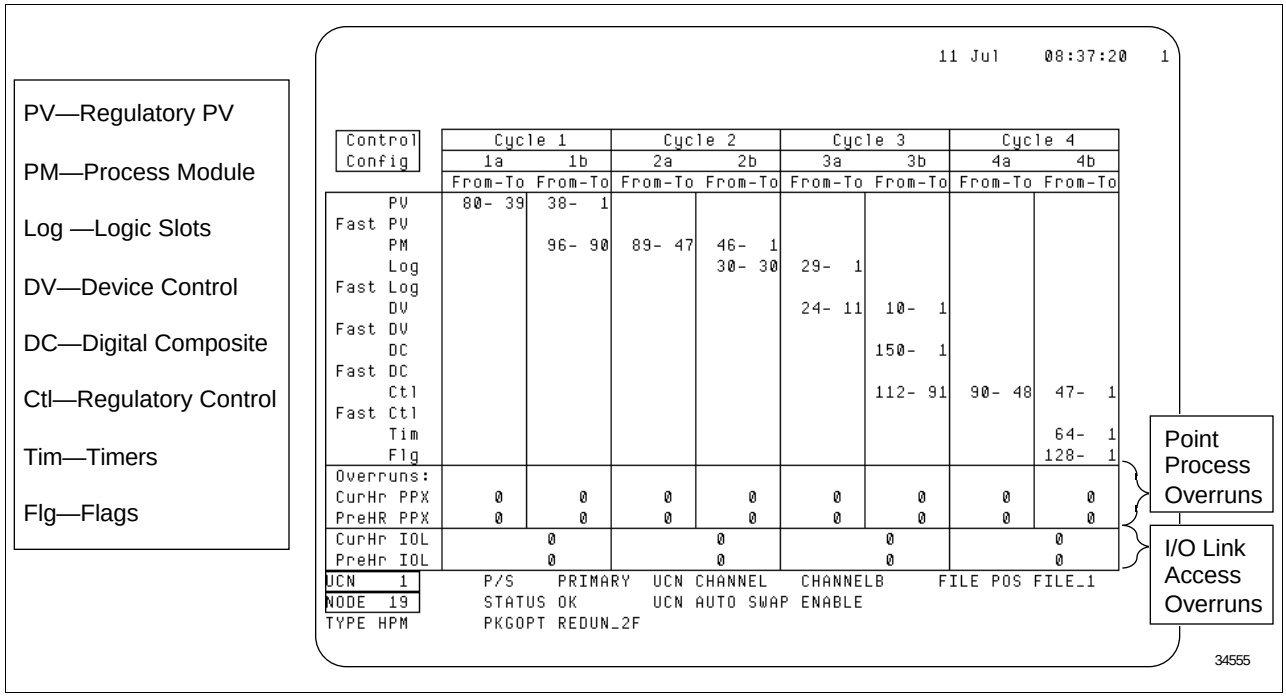
Introduction

Two types of overruns are maintained in the schedule display:

1. Point processing overruns
2. I/O link access overruns

An example of where the point processing (abbreviated as PPX) and I/O Link overrun (abbreviated as IOL) appears is shown in Figure 6.

Figure 6 Schedule Fields



Processing overruns

Point processing overruns occur when

- Points scheduled to be processed during a 1/8-second subcycle cannot be finished within the allotted time. Note that the points do finish execution.
- Point processing overrun soft failures occur if overrun occurs on any subcycle 4 seconds in a row. The current hour overrun counter is incremented by one.

Cause and recovery

The primary cause of point processing overruns is several large sequence programs on the same subcycle. The alarm clears when an overrun does not occur for 8 seconds. Note in Figure 6, for each subcycle the current hour and previous hour overrun counters are maintained in the APMM to help track down the cause of the overrun.

Continued on next page

Identify Overruns, Continued

Definition of previous hour and current hour

The definition of previous hour and current is as follows:

- Current hour statistics represent the counts for the hour you are currently in. That is, if the system time is 12:20 PM, the statistics are accumulated from 12 PM and represent 20 minutes of statistics.
- Previous hour statistics represent the statistics for the previous hour. That is, if the system time is 12:20 PM, the statistics represent the counts from 11 AM until 12 PM.

Illustration of overrun

A simple example may illustrate the relationship of the overrun counter fields that appear in the subcycles to point processing. For example, if there are 160 Regulatory control points, there will be 20 per subcycle (assuming a 1-second scan configured). These points have 125 milliseconds to run. If they take more time, a point processing (PPX) overrun occurs *for that subcycle*.

Experience has shown that the most likely cause of point processing overruns are lengthy programs. For that reason, consider how process module points are executed.

How process module points run

Based on rules of thumb, about 10 CL statements per ProcMod point can be processed in a step per second. If process module points are configured for two processing units (2_PU), on the average, 20 statements per sequence step can be executed. An engineering test with a mix of instructions found that a "typical" instruction takes **.**1PU.

Process module points running in one sub cycle may cause point processing (PPX) overruns for that subcycle, *but not for other subcycles*.

What this means is that you can have a program with 256 statements in one step, but if *all* the sequences on a subcycle have this many statements per step, overruns will occur. As long as all of the sequences on a subcycle average out to 20 statements per step, no overruns will occur.

In practice, many sequences may spend most of their time simply checking for special conditions and not use much time, but then will execute a large number of statements when the checked conditions are met. Consequently, the average number of statements may be within limits, even though large numbers of statements may be executed occasionally.

Conclusion

If there were only one process module point in that node's subcycle and no other points, you could probably have 256 statements in one step and not get any overruns.

Possible Solution

After isolating the program causing overruns, possible solutions include

- placing the program on another cycle, or
- adding pre-emptions to steps in the program.

Continued on next page

Identify Overruns, Continued

I/O Link access overruns

I/O Link access overruns occur when parameter read and write access request, from memory slots to IOPs in the same node, are not completed within a cycle time. Overruns indicates that too many parameter access requests were attempted though the I/O Link within the last cycle (250 msec.). When overruns occur, point processing is delayed by 125 msec. and the I/O Link overrun counter is incremented by 1; this continues until the requested data becomes available.

Cause and clearing

The primary cause of I/O Link overruns is parameter requests larger than 240 for each 250 msec cycle. The soft failure alarm is set if an overrun occurs during any of the 1/4 cycles for 4 seconds in a row. The soft failure clears when an overrun does not occur for 8 seconds.

Possible causes

The Data Entity Builder for memory points (such as logic slots) restricts the number of I/O Link accesses that you can make. Additionally, an overrun occurs only when 200 parameter requests per cycle are exceeded. For these reasons, you should not encounter an I/O Link overrun. In the unlikely event an I/O Link overrun occurs, there are several possible causes:

- Unusually heavy CL/PM poststores to the I/O link requests. Prefetches usually won't cause an overrun, because the program would fail. For the APM and PM, prefetches are limited at 12 per program step.

  One possible solution is to store the data to the memory point instead of the IOP point.
- An unusually heavy LCN request that makes I/O link requests. This could be from a schematic, history group, report, AM, or CG program.

  One possible solution is to request the data from the memory point instead of the IOP point. Another solution is to stagger the requests (for example, an AM requests the data at 60 seconds + a randomly generated number.)
- Extensive use of I/O diagnostics displays.

  A possible solution is to limit the USs that display the diagnostic data.
- Peer UCN requests.

  One possible solution is to request control data rather than I/O Link data. This is because the data already resides in the HPMM/APMM/PMM global memory.
- Excessive Archive Replay Module (ARM) requests.

  One possible solution is to request data on less frequent schedules
- Excessive IOL events during startup.

  View the System Status Change Journal for the number of recorded events exceeded indication (NRECO appears at the end of journal).

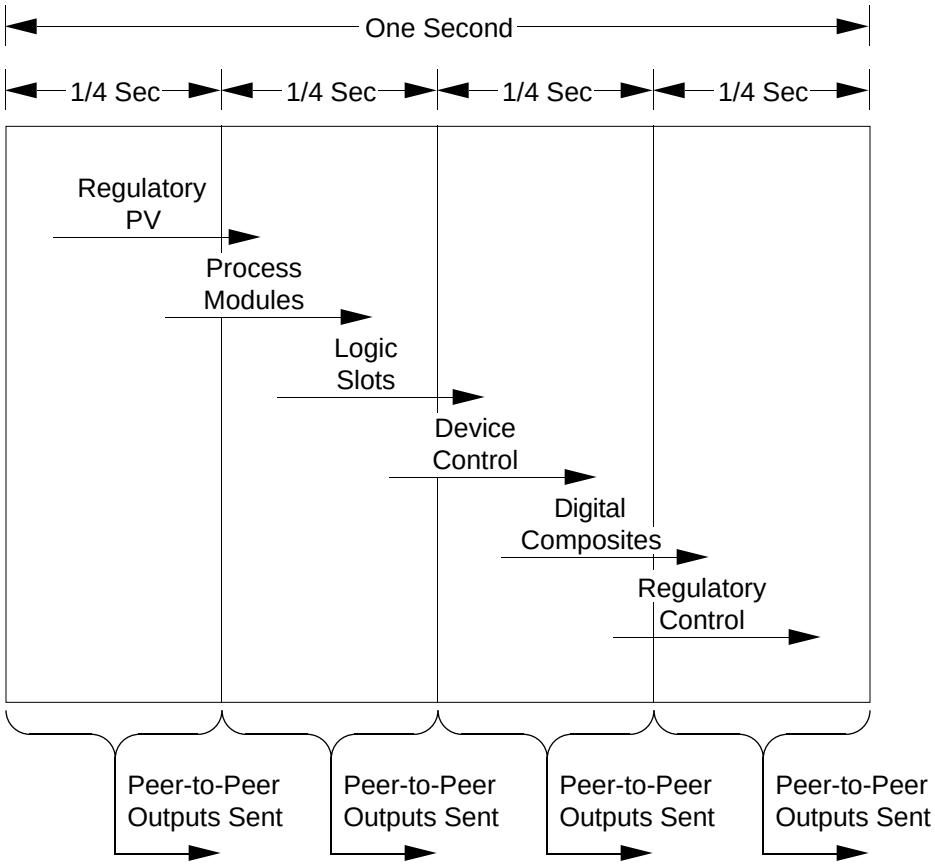
Optimize Peer-to-Peer Communication

Introduction

As discussed in the course material on peer-to-peer communication, the schedule display is also used to determine how many messages are sent when output connections are used. (Recall that if stores or points are sent to the same node destination on a cycle, they are counted as one message.)

Figure 7 shows when outputs to peer connections are sent.

Figure 7 Schedule Peer Outputs



Concept: Peer-to-peer outputs are generated (message sent) at the time the point is executed.

9848

Troubleshoot with the Schedule Display

Introduction

This section shows you how you can use the schedule display for troubleshooting the following:

- Point processing overruns
- I/O Link overruns

Processing overrun troubleshooting

If for some unexpected reason you need to troubleshoot an overrun problem, the schedule display provides information you can use. For example, typically you would

- Identify what cycle(s) the overrun is occurring in.
- Identify the points in that cycle.
  - Call up the Slot Summary display for those points.
  - If Process Module points are in the cycle, identify the programs they are running.
- Get the program names (you can use Documentation Tool to find them).
- Get a printout of the program listings.
- Identify whether programs can have additional steps (or pre-emption points) inserted or moved to different cycles.
- Usually the overrun can be resolved through CL optimization. This includes
  - add pre-emption points
  - Precalculating variables
  - Calling smaller subroutines
  - Adding conditions (IF,THEN) to skip as much code as possible.
- A very unusual CL/AM optimization scenario could include placing random number generators in CL/AM code. This scenario is encountered when several AM programs with parameter requests execute on the same schedule.

I/O Link overrun troubleshooting

If for some unexpected reason you needed to troubleshoot an I/O link overrun problem, check the following:

- LCN requests for I/O-resident parameters. For example, schematic requests to I/O.
- UCN requests from peer nodes for I/O-resident parameters. For example, peer connections to I/O instead of control points.
- Local node requests for I/O-resident parameters. For example, serial interface to foreign devices could request data less frequently.

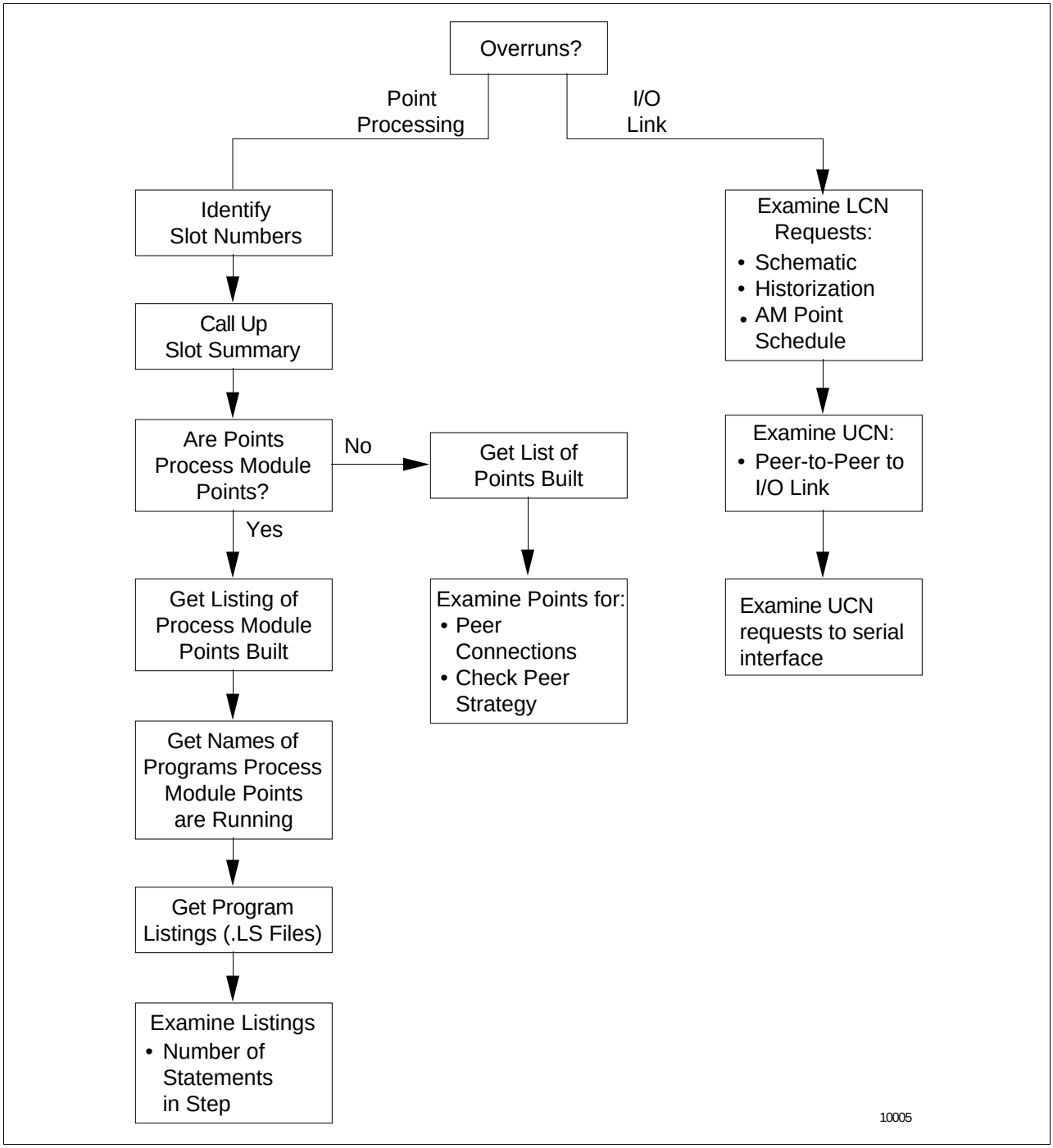
Continued on next page

Troubleshoot with the Schedule Display, Continued

Troubleshooting summary

A troubleshooting approach using the schedule display is summarized in Figure 8.

Figure 8 Troubleshooting Approach



Lab Exercise

Display Orientation

Introduction

The following lab exercise uses the UCN node schedule to identify

- point execution cycles, and
- overrun information.

Lab requirements

- Your course manager may decide to insert various conditions that cause overruns.
- Some lab exercises must be coordinated as a group effort; to maintain lab control only one “bug” is inserted at a time.

Display orientation

Call up the schedule display for your assigned node and become familiar with its layout.

Step	Action
1	Call up the UCN Status display.
2	Select the UCN node assigned to you.
3	Call up the detail display.
4	Select the UCN node assigned to you.
5	Call up the detail display.
6	Select the Control Configuration target.
7	Select the Schedule Info target
8	Identify the following fields: • _____1/4-second cycles • _____1/8-second subcycles • _____Overruns fields • _____Slot execution orders

Identify Overruns

Identify processing overrun

Your course manager will insert a condition that causes a point processing overrun to occur. Locate the overrun error and the condition that caused it.

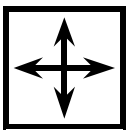
Step	Action
1	Notify your course manager when your work group is ready to have a bug installed in your assigned UCN node.
2	Locate the cycle the overrun occurs in.
3	Call up the appropriate display to identify the points executed in the cycle.
4	Identify the point(s) causing the overrun.
5	Notify your course manager when you feel you have located the bug.
6	Your course manager will then remove the bug and indicate whether or not an additional bug is installed.

Identify I/O Link overrun

Your course manager will insert a condition that causes an I/O Link overrun to occur. Locate the overrun error and the condition that caused it.

Step	Action
1	Notify your course manager when your work group is ready to have a bug installed in your assigned UCN node.
2	Locate the cycle the overrun occurs in.
3	Call up the appropriate display to identify the points executed in the cycle.
4	Identify the cause of the overrun.
5	Notify your course manager when you feel you have located the bug.
6	Your course manager will then remove the bug and indicate whether or not an additional bug is installed.

Directions



DIRECTIONS—This is the end of the study material for this module. Discuss questions concerning the study material or the lab activities with a colleague or a course manager

If you are satisfied that you have achieved the objectives of this module, continue with the next section, the Student Proficiency Evaluation.

Student Proficiency Evaluation

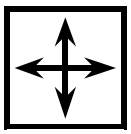
Criterion Test

Test item 1	Describe a scenario where the schedule display would be used to identify a less than optimum control strategy design. You can use an example from the course material, or describe an example you think would not be the best strategy.
Test item 2	Point to the information displayed in the schedule display such as • Slot execution order, • Execution subcycles, and • Overruns.

Self-Evaluation

Test item 1	*Refer to the section “Identify Point Before/After Schedule.*
Test item 2	*Your course manager may ask you to call up the schedule display and point to the listed information.*

Directions



DIRECTIONS—This is the end of this module.

Use your course map to

- Get your course manager to sign off this module.
- Choose your next eligible module.

If you have a question

- Ask your course manager.

LAST PAGE