Index Panel

SLC Control Program

SLC INDEX

SLAC's Software Engineering Newsletter

January 31, 1996 **All that Fits is News to Print** Vol. 9, No. 9

Lattice Diagnostic Pulse Facility

January 23, 1996

Author: *Hendrickson/Underwood* **Subsystem:** *Linac* **User Impact:** *None*
Panel Changes: *Few* **Documents:** *Yes* **Help File:** *Yes*

The Lattice Diagnostic Pulse Facility (LDP) has been added to provide some insight into undesirable changes in the Linac lattice. For example you should be able to monitor diurnal changes in the Linac lattice or identify a mis-phased klystron. This facility has been added to the SCP, and in addition it runs as part of the slow feedback process.

The LDP management touch panel is accessible from the Linac LEM touch panel. The touch panel is initialized to use the NDR_ELEC display group (DGRP) and a sub-range of micros from LI02 through LI28. Both ZPLOT and tabular displays are available for the latest LDP data as well as up to 2 sets of previously saved LDP configuration data.

The LDP data is generated by using 1 or 2 NRTL post kickers to perturb the beam orbit. The unperturbed and perturbed beam orbits are measured and the data is analyzed to fit the betatron oscillations. The betatron oscillation phase advance as well as the 2 kicker phase difference is calculated for each BPM and placed into the BPMS LDP secondary.

The LDP phase advance measurement is currently made by kicking beam pulses destined for the sector 28 off-axis profile monitors. Routine measurements are triggered by slow feedback once per hour and configurations saved once per shift for later analysis. The feedback process is accessed through the new SLOW WATCHDOGS group on the slow feedback panel, and it is the DIAGWTCH loop. Additional details about the acquisition are available in the touch panel help on the slow feedback panel, and on the Lattice Diagnostic Pulse panel.

The user can request a measurement at any time with an option to save configurations.

The touch panel user interface features DGRP and micro range selection, display selections, and configuration management. The user can load 1 or 2 previously saved configurations for comparison with each other or with the latest data. The LDP data may be viewed as ZPLOTs or in a tabular form. Fresh data may be acquired with or without configurations being saved.

The $\boxed{\text{SELECT DGRP}}$ and $\boxed{\text{SELECT MICRO RANGE}}$ buttons determine the range of BPM LDP data that will be displayed. The DGRP and micro range default to NDR_ELEC and LI02 through LI28, but may be changed as required. Changes to these buttons affect only the displayed data and not the acquisition of the data.

The configuration controls are used to select and display previously saved BPMS LDP data. The configuration region is forced to be LDP_DATA and can not be changed. The user may load one or two configurations by pushing the $\boxed{\text{LOAD CNF 1}}$ or $\boxed{\text{LOAD CNF 2}}$ buttons. The user will be prompted for the configuration number to be loaded with the default always being the last configuration number that was saved. Use the $\boxed{\text{DISPLY INDEX}}$ button to determine the desired configuration and enter the number at the prompt. The BPMS LDP data will not be actually loaded back into the database like normal configurations, but will instead be placed into local memory for use by the displays.

The $\boxed{\text{GET FRESH DATA}}$ and $\boxed{\text{SAVE CONFIG}}$ buttons are used to acquire new data. If a configuration is to be saved, the user will be prompted for a configuration title. Since pulses are being stolen at a low rate, the data acquisition will take 3 minutes to make a new measurement. If there are too many bad BPM measurements, configurations will not be saved and an error message will be issued.

The $\boxed{\text{DISPLY TEXT DATA}}$ and $\boxed{\text{DISPLY ZPLOT DATA}}$ buttons will display either tabular or graphic versions of the type selected by the $\boxed{\text{DISPLY TYPE}}$ button. This button will toggle between the permitted types of displays which will depend upon whether 0, 1, or 2 configurations have been loaded. The upper ZPLOT will either display the absolute betatron phase advance for the latest measurement or a saved configuration, or the difference between the latest measurement and one configuration or between two configuratons. The lower ZPLOT display will be shown only if the latest measurement or the configuration contains valid 2 kicker difference data.

MPS Fault Accounting Automation and Enhancements

Jan 23, 1996

Author: *Karey Krauter* **Subsystem:** *MPS Fault Accounting* **User Impact:** *Small*
Panel Changes: *Two panels* **Documents:** *No* **Help File:** *No*

A combination of (1) features added to the mps algorithm compiler and (2) a reorganization of the mps fault accounting touch panel for displays and history buffer plots have resulted in a couple of revised displays and touch panels.

The mps algorithm compiler MPSL and the hand-holding EDITMPS tool have been enhanced to generate an additional output file. The original MPSL PORT and OBJ output files are unaffected; these are still strictly for downloading to the mps AP's for digital signal interpretation. A new MPSL switch has been added called "/DBS" to cause the generation of a fault accounting database file for use in MPS Fault Accounting.

The DBS file generated has the name "FLTS*apXX*.DBS" and is stored in the CMS directory named REF_DBSFILE. There is one file corresponding to each MPS AP for which there is a compiled algorithm. Listed in this file will be a fault accounting entry for every mps device in the algorithm. A device's entry location remains the same from algorithm compile to algorithm compile in order to keep the history of that device/entry valid and continuous across compiles. This will be true even if devices are added or removed or relocated in the algorithm.

Since MPS Fault Accounting dispenses its information based on a device's area and type, this information is also maintained in these new dbs files. As mentioned in this issue's related Index Panel article on the new MPSL DINCS switch used by DSP and SIP displays, these areas and types are limited to those listed in the first few units of REF_DBSFILE:FLTSVX40.DBS. An individual device's area and type assignment is done in that device's module database. That means in the LDIM primary, the RTDP primary, and the PICP primary, there are now new AREA and TYPE secondaries with channels in them up to the number of input bits in each module. Note that areas and types are assigned to input bits, so to prevent ambiguity the same area and type should be assigned to all the input bits used by the same device (this is mainly an LDIM issue). There are 32 area/type channels listed in each mps LDIM, 16 for RTDP's, and 5 for PICP's. Examples of these may be found in REF_DBSFILE:LDIM.DBS, REF_DBSFILE:PIC_AB01.DBS, and REF_DBSFILE:RTD_AB01.DBS. Comments in these files are the only help in keeping straight what device owns what input bit, so have a care that these comments may fall out of date. Unfortunately we couldn't think of any other efficient unintrusive place to put this per-mps-device information.

EDITMPS unconditionally produces this DBS output file and puts it into CMS for you if it has changed from the existing CMS version. EDITMPS now also asks you a couple questions related to keeping the fault accounting database clear of obsolete devices: "MPSFLTCNT_CHECK for spares in flt accting?" means to list how many empty spare entries are still available, and "MPS-FLTCNT_CLEANUP invalid devices in FLTS*apXX*.DBS?" means to cleanup the fault accounting database to create more empty spare entries. CLEANUP is recommended if the CHECK shows fault accounting is getting low on spares. The fault accounting process must be restarted in order to pick up database changes, and EDITMPS also asks if you desire this action too (it's pretty noninvasive).

Note this involved an interesting number of changes to inquiries in EDITMPS. If you have any problem using it, or it leaves remnants behind in your default directory, I would be delighted if you contacted me about it (my email is KEK).

As fallout from all this MPS Fault Accounting spring cleaning and area/type enlightenment, the MPS Fault Accounting Display and History Buffer touch panels have been rearranged.

The MPS Fault Accounting Display touch panel now has dynamic buttons for the area and type filter selection. To display fault accounting trip counts from the MPS Fault Accounting Display touch panel, you must first choose the time interval over which you wish to know about device trips (e.g. last six minutes, hour, shift, day, etc). Then you may optionally limit the listing to a particular device micro (e.g. AB01) and/or a particular device type (e.g. IONCHAMBER) and/or a particular machine area (e.g. 50LINE). Note that all of these optional filters may now be used in conjunction with each other - it didn't used to be this way.

The MPS Fault Accounting History Buffer touch panel has been completely remodeled to be consistent with the Fault Accounting Display touch panel. This touch panel has two main functions: one is a directory function listing device entries in the fault accounting database, the other is to set up to history buffer plot a device's trip history. For doing the directory function, having all the filters available from the Display touch panel will make the display easier to pare down to small bite sizes. The history buffer plot setup must now deal with multiple fault accounting micros - it used to just be VX40 that retained all the fault accounting history, but now VX40 only holds the history for non-MPS devices such as are used in SIP. MPS device trip history is held in the appropriate AP micro name (e.g. AP01, AP02, AP04). Suggested use for the History Buffer touch panel is to get a directory display of the device in question, extract the fault accounting entry identification from the directory display (fault micro, unit, and channel), then enter that fault accounting entry information into the history plot setup buttons and invoke the history plot panel. Note that the fault accounting channel may have to be re-entered once on the actual history plot panel.

MPS Algorithm compiler helps out displays

Jan 23, 1996

Author: *Karey Krauter* **Subsystem:** *Displays* **User Impact:** *Small*
Panel Changes: *None* **Documents:** *No* **Help File:** *No*

A combination of (1) features added to the mps algorithm compiler and (2) a reorganization of the digital and analog status displays for mps devices have resulted in a number of revised displays.

The mps algorithm compiler MPSL and the hand-holding EDITMPS tool have been enhanced to generate additional output files. The original MPSL PORT and OBJ output files are unaffected; these are still strictly for downloading to the mps AP's for digital signal interpretation. A new MPSL switch has been added called "/DINCS_AND_SINCS", or "/DINCS" for short, to cause the generation of output files which may be used as include files in DSP files (thus the name "DSP INCLUDE FILE" or "DINC"). "SINC" files or "SIP INCLUDE FILES" are also generated for use in SIP displays. From EDITMPS, this feature is available by answering "YES" to the question "Do you wants dincs and sincs also?". The SINC files generated are identical to the DINC files generated except for inserting correct SIP syntax where appropriate, and therefore will not be refered to explicitly for the remainder of this discussion.

A number of DINC files are potentially generated, depending on the devices used in the algorithm being compiled. Each MPS device used in an algorithm should have an AREA and a TYPE assigned to them for display organization purposes. The exact list of AREA and TYPE names available for assignment to devices is limited to those used by MPS Fault Accounting, since this DINC feature is tied in with the new MPSL DBS switch used by MPS Fault Accounting (see related Index Panel article this issue). Currently this list is as follows: (as extracted from the first few units in REF_DBSFILE:FLTSVX40.DBS - note: this is all database driven.)

Areas:

INJCLI01	NDRSYS	LI02LI19	SCAVEXTR	POSIEP02
POSRLWTA	SDRSYS	LI20LI30	NARC	SARC
NFF	SFF	FFTB	ESA	PEPHEINJ
PEPLEINJ	50LINE	51LINE	52LINE	COMMON
NFREDIN	NFREDOUT	SFREDIN	SFREDOUT	

Types:

RTD	IONCHMBR	FLWSWTCH	TIU	VACUUM
COLIMATR	RF	OBSTRUCT	MAGNET	TARGET
SUMMARY	WIRESCAN	LDIM		

While the list of areas and types that may be assigned to a device is limited to what is listed in REF_DBSFILE:FLTSVX40.DBS, the actual place where these areas and types are assigned to each device is in each device's module database. That means in the LDIM primary, the RTDP primary, and the PICP primary, there are now new AREA and TYPE secondaries with channels in them up to the number of input bits each module has. Note that areas and types are assigned to input bits, so to prevent ambiguity the same area and type should be assigned to all the input bits used by the same device (this is mainly an LDIM issue). There are 32 area/type channels listed in each mps LDIM, 16 for RTDP's, and 5 for PICP's. Examples of these may be found in REF_DBSFILE:LDIM.DBS, REF_DBSFILE:PIC_AB01.DBS, and REF_DBSFILE:RTD_AB01.DBS. Comments in these files are the only help in keeping straight what device owns what input bit, so have a care that these comments may fall out of date. Unfortunately we couldn't think of any other efficient uninstrusive place to put this per-mps-device information.

A DINC file is generated for every area and type of device used in the MPS algorithm, with a filename of form "*area_type_apXX*.DINC". If more than one device is in this area and type then more than one device will be listed in this DINC file. The devices are listed in these files in the form of DEVSUMY statements so they may be used in DSP displays. There are a small number of devices that are used in their algorithm in such a way as to make them incompatible with this single-area/single-type-per-device assumption; these are devices that should be displayed differently depending on what mode the machine is running in, and these devices will be flagged at algorithm compile time with the warning "!MPSL: FYI: special *area_type* dinc needed for *device*." These special devices will need special handling when their DSP files are constructed. Here is an example of the DINC files automatically generated when compiling the current SLC96 AP01 algorithm:

50LINE_FLWSWTCH_AP01.DINC   50LINE_IONCHMBR_AP01.DINC

50LINE_VACUUM_AP01.DINC
51LINE_FLWSWTCH_AP01.DINC
52LINE_FLWSWTCH_AP01.DINC
COMMON_FLWSWTCH_AP01.DINC
COMMON_OBSTRUCT_AP01.DINC
COMMON_RTD_AP01.DINC
COMMON_VACUUM_AP01.DINC
FFTB_FLWSWTCH_AP01.DINC
FFTB_OBSTRUCT_AP01.DINC
FFTB_SUMMARY_AP01.DINC
FFTB_VACUUM_AP01.DINC
NARC_IONCHMBR_AP01.DINC
SARC_IONCHMBR_AP01.DINC

These DINC files are all stored in CMS in the REF_DSP directory (a.k.a. SLCDSP). The identical SINC files are put into REF_SIP (a.k.a SIP_SOURCE or SIP_CONTROL). EDITMPS will do this for you, for each file it produces that is changed from the version in CMS.

Note this involved an interesting number of changes to inquiries in EDITMPS. If you have any problem using it, or it leaves turds behind in your default directory, I would be delighted if you contacted me about it (my email is KEK).

Of course all these DINC and SINC files are only so much chaff unless some DSP display and SIP display uses them. See the following article in this Index Panel.

MPS Generated Include Files for Summary Displays

Jan 23, 1996

Author: *S. Bes* **Subsystem:** *MPS Summary Displays* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *No* **Help File:** *No*

The ARC MPS, FFTB MPS, BSY/ESA MPS DCXP displays are now made up of include files generated from the MPS algorithim. MPS devices are assigned an AREA and TYPE in an LDIM secondary. When the MPS algorithim is compiled, files are generated that contain devices common to a specific AREA and TYPE. These files have the form: AREA_TYPE_APXX.dinc. The files are maintained in CMS as part of the display (.DSP) library.

This essentially makes the more commonly used MPS displays automatically generated. Hopefully, these displays will now require very little maintenance and should be more accurate in the devices displayed as we switch between the various running modes. Displays will continue to use the standard DCXP format, i.e., select a box and show all or show faulted devices, so this change should for the most part be invisible to the user. The new ARC MPS display is probably the best example of the use of MPS AREA in generating displays. For example, the buttons [50LINE DEVICE SUMMRY] and [COMMON DEVICE SUMMRY]

List all devices in the 50LINE and COMMON areas that are in the current running algorithm; that is they are relevent for SLC running. Some areas are not necessarily geographical, but contain devices that must all be in one state or another for the beam to be able to go to the next geopraphical area. The FRED area buttons comprise one example: NFRED OUT SUMMRY | SFRED OUT SUMMRY | NFRED IN SUMMRY | SFRED IN SUMMRY

The FRED buttons list devices that make up the FREDIN and FREDOUT states. If the boxes corresponding to the FREDIN states are green (all devices are made up), then the beam may not proceed down the ARCS and any faulted devices in the ARCS would not cause an MPS trip. Likewise, if the FREDOUT state is made up, the beam may proceed down the ARCS, and faulted devices in the NARC and SARC would cause an MPS trip. To avoid confusion, we informally referred to AREAs such as this as area delimiters. The SL2's and 50B1 are other examples of delimiters.

Other displays, such as the FFTB MPS display are in of themselves very area specific, and therefore rely on the TYPE designation for display formatting. The types used in this display, and most commonly in other displays are TEMP, IONCHMBR, and LDIM.

The SIP MPS summary display also makes use of these include files in the regions BSY, FFTB, ESA and in the beginning of the ARCS. Summary includes have the file format and type:

AREA_TYPE_APXX.sinc

and are maintained in the SIP library.

It should be noted that files of the type .DINC and/or .SINC can, but SHOULD NOT be edited with editdsp, editsip or through CMS reservation. The files would then not be consistent across algorithm compiles. Any problems pertaining to files of this type should be refered to the database group.

BITBUS Network - PEPII magnet support

January 10, 1996

Author: *Lee Ann Yasukawa* **Subsystem:** *Magnets* **User Impact:** *Large*
Panel Changes: *None* **Documents:** *Yes* **Help File:** *Yes*

The PEPII magnets (rings and very last part of the injection line) will be controlled by a new power supply controller over a new network. This setup is called a BITBUS network and was developed by INTEL.

The actual power supply controllers (which contain the DAC, ADC, and regulation circuitry) are being built by the power conversion group. Each PEP-II micro will have a new (commercially produced) multibus BITBUS controller card installed in it. The micro will submit requests to the controller over the BITBUS network and receive responses once the requests have been performed.

The functionality of the magnet software will not change just the communication mechanism - the new network will talk BITBUS instead of CAMAC.

To support this new network, bits and pieces of software have been released recently with more to follow. The pieces of software which have been released so far are summarized below.

1. A new micro subjob was created to accept and process BITBUS requests from the SCP or VAX based standalone processes. This subjob is similar to the Virtual CAMAC job on the micro which can perform CAMAC operations to CAMAC modules controlling magnets independent of the magnet subjob.

   This BCOM (BITBUS Communication) subjob is up and running in the SLC micros and waiting for BITBUS requests (which will not occur for SLC).

2. A new VAX based diagnostic tool has been created to communicate with the BITBUS networks via VAX DCL. This tool is similar to CAMCOM for CAMAC operations but was actually modelled after the MCOM Diagnostic Tool for KISNET. It has the same "look and feel" of the MCOM Diagnostic Tool.

   This BCOM Diagnostic Tool is installed on the MCC and SLC VAXes and can be invoked by typing BCOM at the DCL prompt. (You may have to logout and login to get the BCOM symbol definition). This tool has on-line help available by typing HELP at the BCOM prompt (HELP is currently not up-to-date but will be soon).

3. In order to support additional status bits from the micro for magnets, a new STA2 database word was added (since the STAT word is almost full). There are a few bits which are already defined although no software updates it yet. The state of these status bits can be seen with the SCP Decode Status Bits display on pages 2 and 3 by using the Next Page and Prev Page buttons.

The software modifications which are in progress, but not yet released, to support the new BITBUS network are summarized below.

1. The micro magnet subjob is being modified to support submitting requests over the BITBUS network. All the applicable current magnet functions will be supported as well as new functions specific to PEPII.

2. There will be a new SCP BITBUS network display to show the status of each BITBUS controller.

3. There will be a new SCP diagnostic display for all BITBUS controlled magnets to relate all diagnostic information available to the VAX about each BITBUS power supply controller.

4. A new PEPII RAMPing function will be supported which will ramp the magnetic fields for groups of magnets simultaneouly.

Additional Index panel articles will be written in the future which will describe the BITBUS network in more detail.

Collimator Emittance Display

January 19, 1996

Author: *Ron Chestnut* **Subsystem:** *Accelerator* **User Impact:** *Small*
Panel Changes: *Few* **Documents:** *Yes* **Help File:** *Yes*

A new SLC linac collimator graphical display as specified by Paul Emma is now available, controlled by the COLL EMIT panel, reachable from the LINAC Collimator panel as well as the LI29 and LI30 Collimator panels. A single linac collimator is choosen as a map point. Then the beam as measured at the sector 28 wires and all LI29 and LI30 collimators are mapped to that point and displayed in phase space. Four seperate phase space displays are generated, electrons and positrons in X and Y respectively.

Additionally a new collimator centering calibration is provided, which averages the current jaw settings and stores that as a center offset. The use of this offset in the display can be toggled on and off via the COLL EMIT panel.

Of interest is the interface to MATLAB used in this implementation. The SCP panel code determines display scales and fetches model parameters, which are all written in MATLAB format to a temporary file, SYS$SCRATCH:COLL_DISP.MAT. Then MATLAB is spawned and automatically runs MATLAB_SLC:COLL_EMIT_DISP.M in the DECTERM. The MATLAB display then remains on the X-display until "EXIT" is entered on the MATLAB DECTERM. Upon return to the SCP code, the temporary file is deleted. The MATLAB user program is automatically started dependent on code found in MATLAB_SLC:MY_STARTUP.M, which runs the emittance code if the temporary file created by the SCP code is present. If a better way can be found, I'll gladly use it.

Scp Release

January 9, 1996

Author: *Ralph Johnson* **Subsystem:** *Any* **User Impact:** *Some*
Panel Changes: *None* **Documents:** *No* **Help File:** *No*

The latest SCP release has the following new features:

- For COWs: If you want to assign knobs 4-7 using a touchpanel having buttons only for knobs 0-3, you can hold down the shift key and push any of the assign buttons to enter into a dialog through which you can enter any knob number.

- There is a new rapid display update feature to avoid having to repeatedly push a display button while waiting for a status update, for example. It is, of course, only useful for displays where the data is asynchronously updated or those for which the data is actively fetched. Please use this for the latter type sparingly, since it will incur high network use. To use this feature, use the F6 keyboard key to toggle rapid display updating on and off. When on, it causes the current graphics display to update at an interval of one second until toggled off or until 30 seconds has elapsed. Some displays, such as klystron displays, cannot yet operate in this mode. If you attempt to turn on rapid updating for a display which cannot operate this way, you will get a message telling you that rapid updating is not available for that display.