THE SLAC BEAM SWITCHYARD CONTROL COMPUTER[*]

S. K. Howry, R. Scholl, E. J. Seppi,
M. Hu, D. Neet[+]

Stanford Linear Accelerator Center
Stanford, California

Summary

A computer system has been developed at SLAC in the beam switchyard for on-line beam transport control. The system consists of an SDS '925' computer and various specially designed interface devices. The system scans and records changes in interlocks and status signals; sets up, monitors, and maintains magnet currents and controls collimator and slit apertures. In addition to standard card and teletype input-output, the system has special panels for operator control of magnet currents and slit and collimator settings. A link to another computer provides for remote information access and control for experimenters. An important feature of the system is the operator oriented language which is provided for operator/computer interaction. The language is arranged so that an inadvertent action of the operator cannot adversely affect the computer program. A single computer instruction can vary a group of magnets simultaneously to change the beam energy while maintaining a given beam focussing condition. A fine-tune control has been provided so that the current in an arbitrary group can be controlled by a single switch. The operator can write instructions in the language which are synchronized in time to 1/360 of a second, and/or repeated at prescribed time intervals.

Introduction

The advent of multilevel interrupts has made the small computer a suitable device for on-line analysis and control of physical data and such systems have been developed in the past. However until recently little attention has been given to the man-machine interface, that part of the system referred to as the command post. This paper describes a system in the beam switchyard at SLAC where this area has been more fully explored. The system is centered around a compact, expandable user language, easily grasped by the operators at the control room, which still provides a high degree of interaction with the environment. All input-output including signals to and responses from the environment are completely multiprocessed, so that the computer is never waiting for a device.

The SLAC beam switchyard is described in detail elsewhere in these proceedings.[1] The principle function of the computer system is to monitor and control the magnet and slits in this area, and to provide certain control capability to the experimenter on a more flexible basis than can be made over a telephone or conventional wired controls. Other functions include data logging and status monitoring. The first section of the paper describes the system components. Later sections describe the language and how the operator uses it to effect the various functions.

System Description

The overall system is shown in Figure 1.

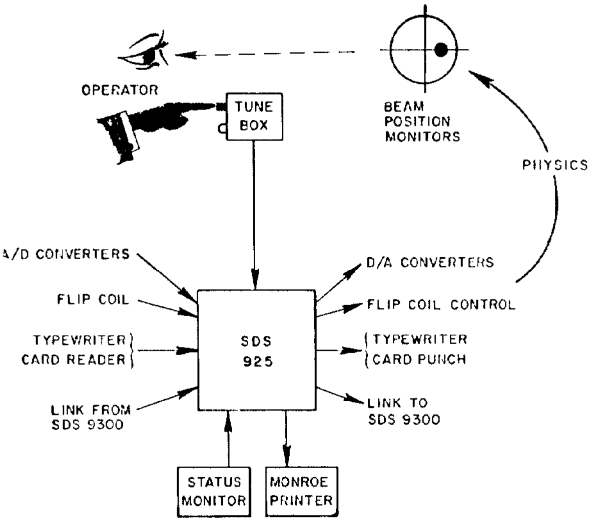


Fig. 1 - Beam Switchyard computer system.

The system includes several pieces of hardware specifically designed for the beam switchyard to control magnets, slits and collimators and to provide man-machine communications. The electronic details are described elsewhere in these proceedings[2] and will only be touched upon here.

The computer is an SDS 925, a 24 bit machine with a 1.74 μsec memory cycle time, and 4K memory. There is a typewriter (teletype Model 35), card reader and punch, all on a single channel, and 16 levels of priority interrupt.

Magnet control is accomplished by providing the magnet power supply with a reference voltage, adjustable by the computer. Most magnets are controlled by D/A converters (rate 1 per 5 msec). For the momentum defining magnets precision considerations require a special system, wherein the computer produces a number which is counted down by a stepping motor, each step creating a fixed current change.

Current is read back by one of two systems, one reading DC signals to 0.005% and one reading the pulsed magnet signals to 0.1%. At present there are 25 channels on the DC multiplexer and 16 on the pulsed system. Approximate rates are 5 channels/sec for the DC and 360 channels/sec for the pulsed readout. Again the momentum defining magnets are treated as a special case; the loop is closed around these magnets through the use of a long coil integrator. This input works

[*] Work supported by the U. S. Atomic Energy Commission

[+] Now at CERN, Geneva, Switzerland

as follows: The computer can command a long coil, mounted in a reference magnet, to make a single 180° "flip". The coil is longer than the effective length of the magnet, and its output is integrated to high accuracy by a voltage-to-frequency converter (0.01%). The result, which the computer can read into its memory is the integral of the field in the magnet along the length of the magnet and is used to calculate the momentum setting of the transport system (see the section on Momentum Definition below).

To allow small changes in magnet currents to be made by the operator with the computer, two "magnet selector boxes" have been provided. The lever on the magnet selector box allows the operator to make a fine adjustment on magnet settings much as he would with a potentiometer. This box is shown in Figure 2. Using the typewriter, he can specify the sensitivity of the lever (TUNE ALL BY .01; gives one percent of value sensitivity, see Table 1 below). Then by pushing the desired magnet buttons and the lever, he can change the magnets independently or in concert. The RATE dial provides single step mode or 3 nominal rates of change.

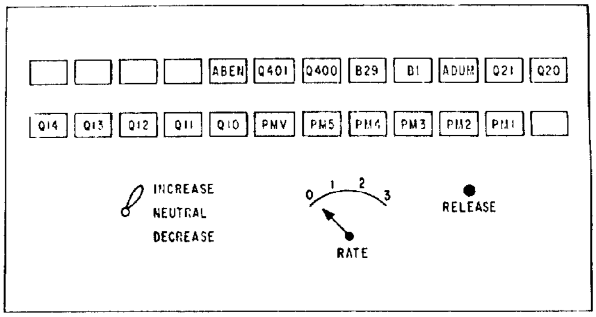


Fig. 2 - Front panel of magnet selector box.

The link with the experimental area A computer (SDS 9300) consists of two 24 bit buffers (one for each direction) along with two interrupt signals to each computer indicating change of state of the registers (full to empty or vice versa). The maximum transfer rates between the two systems is about 10^3 words/sec, limited by programming considerations. The experimental area A computer system is described in reference[3].

The status monitor interface packs status signals (1 for fault condition, 0 otherwise) into 63 sixteen bit words. These are read into the computer 360 times/sec and checked for change of state. Any such change is logged (with the time) on the Model 104C Monroe Printer (a 20 line/sec, digits only 12 char/line device). The information is particularly useful in the diagnosis of intermittent malfunctions detected by the interlock system, which can cause a status signal to trip off for only a fraction of a second. This independent activity is multiprocessed with the other functions, and takes about 50% of the computer time.

User Language

The instructions available to the user are in Table 1. Their full description and programming details are in the references [4, 5]. Language input may come from the typewriter, cards, or over the link from experimental area A. The appropriate 'TO' clause sends output to the typewriter, card punch, or over the link to experimental area A. The typewriter/card channel and the link may be operating simultaneously.

The entities considered in this discussion of the language are: characters, words, clauses, instructions, and blocks. Universal delimiting characters are blanks and carriage returns. These two are functionally equivalent and may be used interchangeably to format the input copy. Words are then strings which contain no delimiters. A clause is a group of consecutive words, one of which identifies the clause.

Thus

```
        L1   : SET B1 = 7.02
```

consists of the clauses

```
        L1   :
        SET B1
        = 7.02
```

The order of the clauses is irrelevant but within a clause position is important. Instructions consist of certain combinations of clauses. The above three clauses make up a legal instruction where the SET and = clauses are necessary but the other one is optional. However, the final nature of the instruction is not decided until the terminating semicolon word. On receiving this the computer searches backwards (i.e., from right to left) for a meaningful instruction. If there is none, the indication

```
        ERR
```

is typed and the accumulated words are cleared out. If there are several, the computer takes the last one. For example, a bad typist might create

```
        STIP STEP Q10 BY .01 STEP Q11 ;
```

and the computer picks out

```
        STEP 11 BY .01  ;
```

In this way the user can always reverse himself in a natural way without special 'word delete' and 'line delete' characters. When educating users, the exact level of permissiveness of the language need not be spelled out as they discover this as they become more experienced.

The optional label clause allows one to give an instruction a name, principally so that it can be de-activated later if desired by a KILL instruction. There is at present no provision for 'cliche' definitions, i.e. typing a label to a block of instructions. However the same effect may be achieved by using card decks for memory as

Table 1

Summary of Instructions

Repeatable Instructions - clauses may be added

```
STEP <elt> BY <no.> ;
CARDS ;
OUTPUT <elt> ;
RECORD <elt> ;
RECORD <elt> ;
SEND <string> ;
TIME ;
CLOCK ;
DATE <string> ;
TOLSCAN <elt> ERROR <no.> ;
```

Optional Clauses

```
EVERY <time>
AT    <time>   -  time clauses
UNTIL <time>
TO <destination> - destination clause
<label> :        - label clause
```

'Once only' instructions - clauses are ignored

```
KILL <label> ;
SET <elt> = <no.> ;
TUNE <elt> BY <no.> ;
TUNE ALL BY <no.> ;
SCALE <elt> BY <no.> ;
```

<time> format examples

```
(3HR)                  refers to
(3:00MIN)              3 o:clock,
(3:00:00SEC)           or 3 hours
(9:00:00:000PUL)
(T+5MIN)               refers to current time
(T+5:00SEC)              plus 5 minutes
```

<destination> list

2 - Typewriter output
3 - Card output
4 - Link output

anything else - typewriter output
no destination clause - typewriter output

<string> definition - any sequence of characters, not spaces or carriage returns, the first of which is a letter or the character '*'

<label> definition - any string

<no.> definition - any sequence of digits with one decimal point somewhere in it. The sequence may be preceded by a minus sign '-'

<elt> list

PM1A	Q10	Q21	ABEN	COV	C10V
PM2A	Q11	ADUM	EF	C1H	T10H
PM3A	Q12	B1	SL10	C1HR	T10V
PM4A	Q13	B29	S10R	C1V	SL30
PM5A	Q14	Q400	SL11	C1VR	S30R
PMV	Q20	Q401	COH	C10H	SL31
BOX1					

the example below shows. The following instructions, on cards, are placed in the reader.

```
LO : EVERY (5MIN) CARDS ;
     (Block 1)
         V
     (Block 2)
         V
     .   .
     .   .
     .   .
     (Block N)
         V
     KILL LO ; V
```

The user types 'CARDS'; this shifts control from the typewriter to the card reader and the instruction LO plus those in block 1 are loaded and interpreted. The 'V' returns control to the typewriter. At the end of 5 minutes LO is again executed and control passes to the reader for block 2. This continues until the last block, along with the 'KILL LO ; ' is read in. This last instruction terminates the activity.

Notice from Table 1 that there is a good deal of emphasis on time. Internal computer time is incremented once each beam pulse (2.8 msec), and this is correlated with time of day. When using the EVERY, AT or UNTIL clauses, the user can synchronize events to a resolution of 1/360 sec over a 6 hour interval. Internally the clock is used to create various real-time delays, so that the external devices may respond. Suppose at time that a delay Δt is required before a number is available for some calculation. The remainder of this calculation is placed in a waiting queue, along with the time $t+\Delta t$, and the computer goes to other work. When internal time is equal to $t+\Delta t$ the number is available and the calculation is continued.

Applications

Computer Magnet Set-Up Procedure

The computer system enables the operator to quickly change switchyard magnet focussing and/or energy acceptance. The former is controlled by a single energy independent number for each quadrupole. These are included in a 'beam focussing deck' as shown below.

```
SEND 'PARALLEL.TO.POINT.FOCUSSING.AT.TARGET' TO 2;
SCALE Q10 BY .0104 KG/GEV ;
.
.
.
SCALE Q21 BY .00781 KG/GEV ;
```

The physicist may either use one of the commonly used decks available on file, or he may create his own from the results of his beam optics studies. The focussing deck is read in and magnets activated by pushing buttons on the magnet selector box. Then the defining energy is given by the instruction

```
SET BOX1 = 7.450 GEV ;
```

All magnet currents are adjusted, closed loop, so that the specified focussing occurs for this energy. Until a new focussing deck is loaded, SET instructions change only the energy definition; quad currents are adjusted to maintain the focussing conditions.

Closed loop control was found to be necessary because of long term drifts in the quantity

$$R = \frac{\text{current delivered by the power supply}}{\text{D/A reference voltage to the power supply}}$$

Although this ratio is never a function of current (i.e. the power supply responds linearly), it was found that component drifts changed its value from week to week. Continuing maintenance work on and around the power supplies also contributed to this problem. Closed loop control is effected by first reading into the computer a number proportional to magnet current. The pole tip field (in KG) is computed using a polynomial obtained from magnetic measurements on the magnet. This is compared with the desired pole tip field as computed from the SCALE number and defining energy. If certain smoothness and loop response requirements are satisfied, the difference Δ is converted to D/A units and dispatched. The loop activity is then placed in a waiting queue for a time proportional to Δ, while the magnet responds. This continues until convergence or until too many iterations occur. There is one such loop for each magnet in the system and all operate independently and simultaneously.

Momentum Defining Loop

The strength of the magnetic field in the large bending magnets leading to experimental area A determines the momentum of those electrons which pass through a narrow slit, and continue to ESA. (See Figure 3). Some experiments call for a very accurate estimate of this momentum. Others require that a given momentum achieved on one day be accurately reproduced later. The operators has at his disposal a current control, but because of hysteresis effects, current does not define the magnetic field (hence the beam momentum) uniquely unless the magnets are first properly deguassed. A slowly turning long coil (flip coil) positioned in a reference magnet measures the field directly to .01%. When a single flip is completed the flip coil interface interrupts the computer with this number and, using calibration data plus switchyard geometry, the computer obtains the momentum of the particles passing through the center of the slit, i.e. the observed setting of the switchyard momentum analyzing system. The computer then adjusts the current to achieve the desired value for this quantity. The response of the bending magnets to several momentum change requests is shown in Figure 4. The above calculation takes very little of the computer's time and other activities may go on simultaneously.

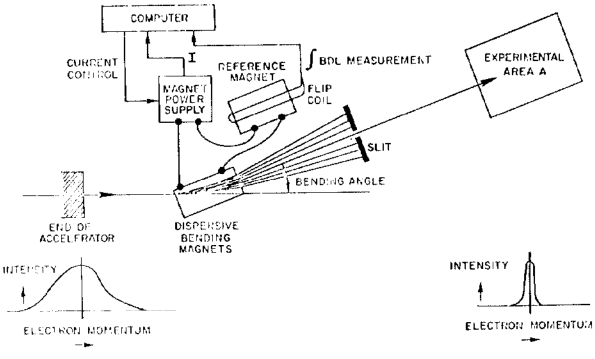


Fig. 3 - Schematic of switchyard momentum definition system.

It is planned that magnet currents and fields will be monitored automatically during experimental runs. This will be accomplished as follows: Instructions like

```
EVERY (20 SEC) TOLSCAN Q10 ERROR .001 ;
```

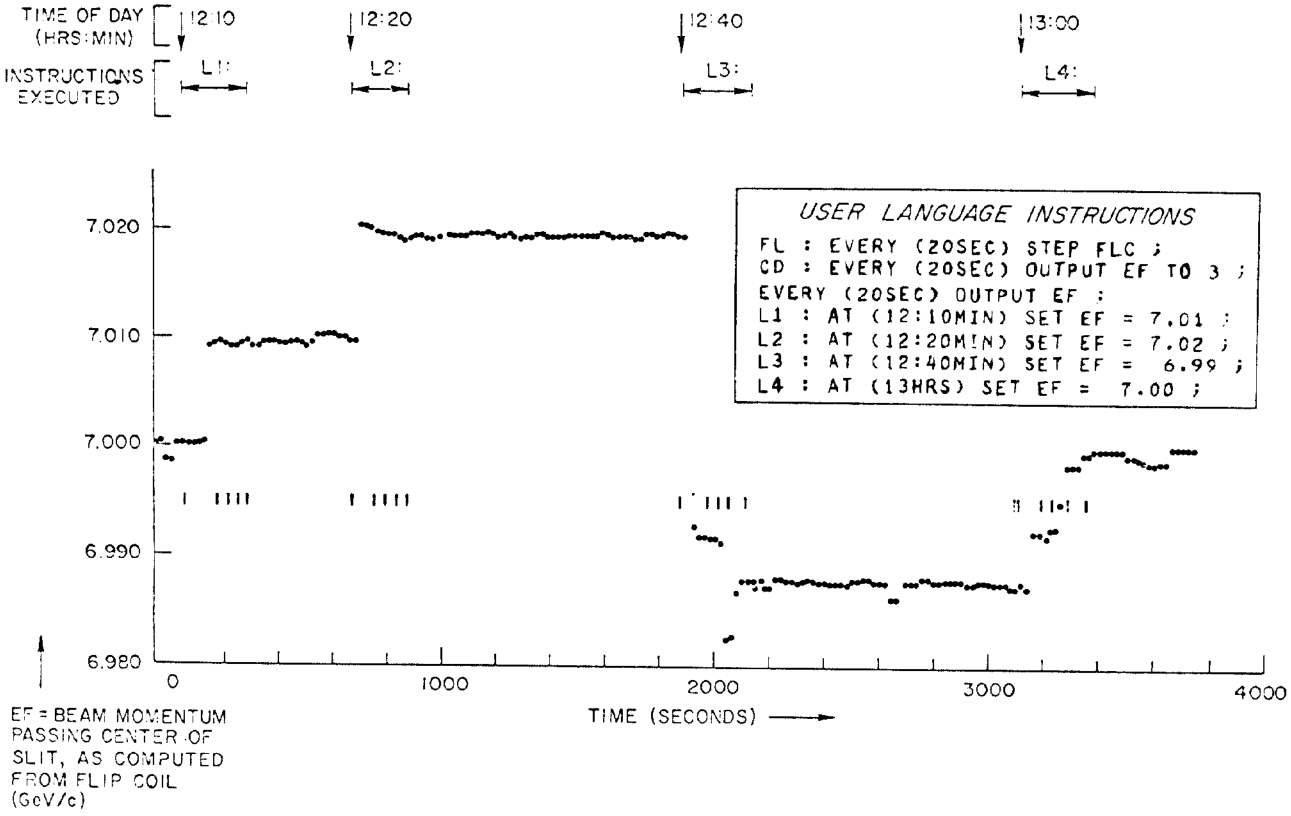
will be included, along with the beam focussing deck described above. At the specified frequency, this instruction will compare the input just sampled with the nominal value. If the tolerance is exceeded, an alarm printout occurs. The test is automatically suppressed if the magnet has recently received a new nominal value.

Calibration of Magnet Loops

Using the source language, it is possible to perform an automatic calibration experiment. The instructions:

```
SETI : EVERY (15 sec) UNTIL (T+10MIN) STEP Q10
                    BY .05 ;
PUNI : AT(T+10SEC)
            EVERY (15 sec) UNTIL (T+10:10SEC)
             OUTPUT Q10 TO 3;
ZERI ; AT (T + 11 MIN) SET Q10 = 0. ;
```

SETI causes the D/A reference voltage controlling the magnet Q10 to be increased by 5% of its value (open loop) every 15 seconds until 10 minutes has elapsed. PUNI cause the number proportional to magnet current to be punched on cards every 15 seconds but 10 seconds behind the execution of SETI. This allows the magnet circuit to complete its response. ZERI returns the magnet to zero current after the run. Similar instructions for all the magnets make up the full calibration deck. Since instructions for different magnets are executed concurrently, the time required is that for a single magnet. Output cards are more conveniently analyzed on a larger computer and results tabulated. Since the operation is automated, it may be easily done on a regularly scheduled basis, and provides a comprehensive maintenance test on the overall system. This example indicates the ease with which different operations can be performed with no programming modifications. In general the system program has been written so that expansion of the instruction list, when necessary, is a relatively simple task.



```
FL : EVERY (20SEC) STEP FLC ;
CD : EVERY (20SEC) OUTPUT EF TO 3 ;
EVERY (20SEC) OUTPUT EF :
L1 : AT (12:10MIN) SET EF = 7.01 ;
L2 : AT (12:20MIN) SET EF = 7.02 ;
L3 : AT (12:40MIN) SET EF =  6.99 ;
L4 : AT (13HRS) SET EF =  7.00 ;
```

Fig. 4 - Magnet response to SET instructions.

Acknowledgements

We wish to thank R. E. Taylor whose advice was invaluable during the early stages of the project. Also the work of Lewis Via who built and debugged most of the hardware and cabling is gratefully acknowledged.

References

[1] H. Weidner, E. Seppi, J. Harris, "Design, Construction, and Early Operating Experience of SLAC Beam Switchyard and Experimental Areas", paper number H-2, this conference.

[2] R. Scholl, R. Coombes, J. Hall, D. Neet, and D. Olsen, "Instrumentation and Electronics for the SLAC Beam Switchyard", paper number I-9, this conference.

[3] R. M. Brown, Mary Anne Fisherkeller, Anthony E. Gromme, John V. Levy, "The SLAC High-Energy Spectrometer Data Processing System", IEEE Proceedings Computer Issue, December 1966.

[4] S. K. Howry, SLAC Report CGTM 10, "BSY Control Computer System Language", August 1966.

[5] S. K. Howry, SLAC PUB-248, "A Concise On-Line Control System", October 1966.