# BUCKET SELECTOR SYSTEM FOR KEKB

E. Kikutani, M. Suetake and M. Tobiyama
KEK, Oho 1-1 Tsukuba, Ibaraki 305-0801, Japan

*Abstract*

A bucket selector system has been developed for the KEK-B collider. Three VME computers for actual bucket-selecting, bunch-current reading and operator interface are working. The system provides a tool for generating the desired filling patterns in the rings.

## 1 INTRODUCTION

At KEK in Japan, the commissioning of an asymmetric electron-positron collider, KEKB, is now in progress. The collider consists of a 3.5 GeV positron storage ring (LER) and an 8.0 GeV electron storage ring (HER). In order to achieve the target luminosity the currents to be stored in the rings are 2.6A and 1.1A for LER and HER, respectively. These beam currents are distributed to many rf buckets and ultimately, the number of buckets filled will be 4607, which is 90% of the total number of rf buckets, namely 5120. At the present stage[1], the maximum storable currents are about 500mA for both rings and the shortest bunch spacing with which beams are stably stored, with the help of feedback systems, is 4ns.

The electron and positron beams are provided by an injector linac complex[2]. For the synchronized operation of the linac and the rings, a fast timing-manipulator system was constructed, and as a sub-part of this system, the bucket selector system was developed. With this system, we can inject the beam into any desired rf bucket of the rings.

In the previous paper[3], we described the basic design of the system. In this paper, we will explain mainly the software structure of this system.

## 2 FUNCTIONS OF THE BUCKET SELECTOR SYSTEM

### *2.1 Generation of desired filling patterns*

In general, it is not easy to store a large number of bunches with a small bunch-spacing. The difficulty arises from various causes: instabilities (mainly coupled-bunch); heating of the beampipes; degradation of vacuum caused by high beam-current; etc. The strengths of the instability, the heating, etc. are functions not only of the total beam-current but also of the bunch spacing, bunch-currents, etc. Thus, a filling pattern of bunches, *i.e.*, which rf buckets are filled with the beam, is a very important factor in raising the luminosity as high as possible.

In principle, the number of possible filling patterns is very large and we can not explore all these patterns. Usually the bunches are filled in trains, and it is convenient to categorize the filling patterns by the following parameters: (1) the number of the bunch trains, (2) the number of bunches per train and (3) the bunch spacing in a train. By surveying these parameters, we find the optimum filling pattern under a given operation condition. Figure 1 shows an example filling pattern: 32 bunch-trains, 40 bunches/train and 6ns bunch spacing.

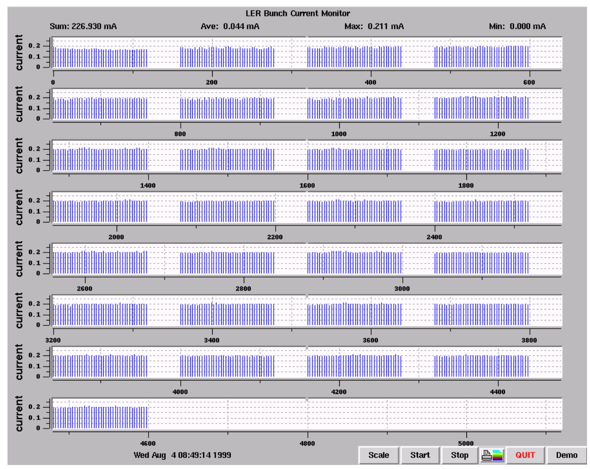


Figure 1: An example of the filling patterns in LER.

The optimum filling pattern may change with different various operation conditions, decrease of the vacuum pressure, change in the strength of the instability depending on the operating point, etc. Thus, it is important for operators to be able to change the filling patterns very easily.

### *2.2 Control of filling order*

Next we explain the requirements on the filling order. If we inject the beam into the same rf bucket continuously at the maximum injection rate (50Hz), the bunches in the buckets surrounding the injection-bucket will be kicked repeatedly. Since the transverse radiation damping time of the rings is about 45 ms, the kicks will be given to the bunches within one-half the damping time. In order to avoid this, the injection-bucket should be changed from pulse to pulse.

### *2.3 Equalizing the bunch-currents in the ring*

When the beam is lost, either slowly due to vacuum or rapidly due to instabilities, we re-fill the ring. It is typical that the bunch currents are lost unevenly and the number of particles to refill is different, bucket to bucket. In order to regulate the bunch-currents we have an optional *bunch current equalizer* (BCE) in our bucket selector system.

## 3 HARDWARE CONFIGURATION

The rf bucket to be filled with the next linac-pulse is determined by the trigger timing for the electron gun. This mechanism is essentially common in both electron and positron modes. Five VME-packaged delay modules are used for controlling this timing, and the delay values of these modules are set by a VME computer which is installed in the common VME sub-rack to the delay modules. Hereafter, we call this computer the "*linac computer*" because the sub-rack is located near the gun.

An accelerator operator chooses the filling pattern (actually, chooses the parameters explained in Section 2.1). These parameters are accepted by a computer installed in the Central Control Room (the "*CCR computer*") . The CCR computer and the linac computer are connected by dedicated optical fiber cables using the interface of the Shared Memory System, which we described in a previous paper[3].

Additionally, we have one more VME computer to control two bunch-current monitors for the LER and HER. They output the 5120 bytes of data per ring synchronized with a signal which triggers the injection kicker. This computer, the "*bunch-current computer*" is also connected to the linac computer with the optical cables and the bunch-current information is transferred to the linac computer.

Figure 2 shows the hardware configuration of the system. The distance between the bunch-current monitor and the CCR is about 300m and that between the CCR and the linac is about 1000m. By using dedicated optical cables, the transfer speed of the bunch-current and the filling pattern information is not dependent on the control network under which all the control components are supported.

## 4 SOFTWARE

### 4.1 General

The control system of the KEKB accelerators, including the injector, has been constructed based on the EPICS environment[5]. In this environment, each piece of hardware is controlled by a VME computer, while the operator interface is performed by UNIX computers. The VME computer is Motorola Power PC 750 (266MHz clock) with memory of 64Mbytes. The operating system is VxWorks 5.3.1.

The program running in the linac computer plays a main role in executing the functions explained in Section 2. In designing the software, the most important point is the regulation of the execution time of the program. The maximum injection rate of KEKB is 50Hz, that is, all the procedure should complete within 20ms. In order to keep the execution time constant, we adopted a scheme as follows:

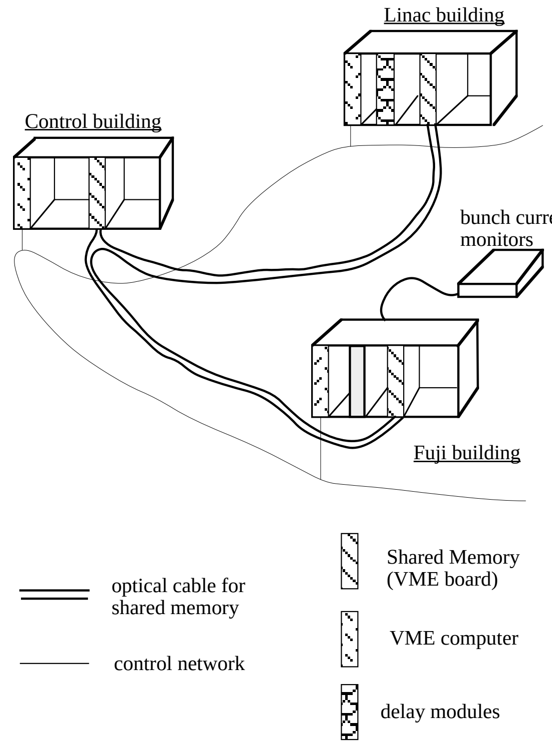


Figure 2: Hardware configuration of the bucket selector system.

1. The number of EPICS records is limited in the linac computer to make it approximately dedicated to the bucket selecting,
2. communication with the operators, which is asynchronous with the beam-injection, is done by the CCR computer to keep the linac computer free from operators' interrupts.

### 4.2 Software on the bunch-current computer

The program in the bunch-current computer acquires the 10240 bytes (two rings) of bunch-current data from the monitors. Its execution speed is fast enough to synchronize to the injection kicker, whose rate is 50 Hz at the highest. It is coded as a *device-support* of the EPICS system and the data is available through an EPICS record. Besides this record, the bunch-current information is written into the Shared Memory. After writing all of the bunch currents, an interrupt signal is also sent to the linac computer through the Shared Memory System.

### 4.3 Software on the CCR computer

As described above, the parameters specifying the filling pattern are accepted by the CCR computer through a number of EPICS records. They are written into the Shared Memory and copies are available on the Memory Board under control of the linac computer. The device support for the Shared Memory loaded in the CCR computer functions as the device support of the bucket selector system.

The operator interface was developed with MEDM, which is the standard user interface in the EPICS system. Figure 3 shows the operator-interface panel. Through this panel, an operator can specify the filling pattern and several options, such as turning the bunch current equalizer on and off. In addition, the operator can read some useful information: the last-filled bucket; total number of bunches; and so forth.

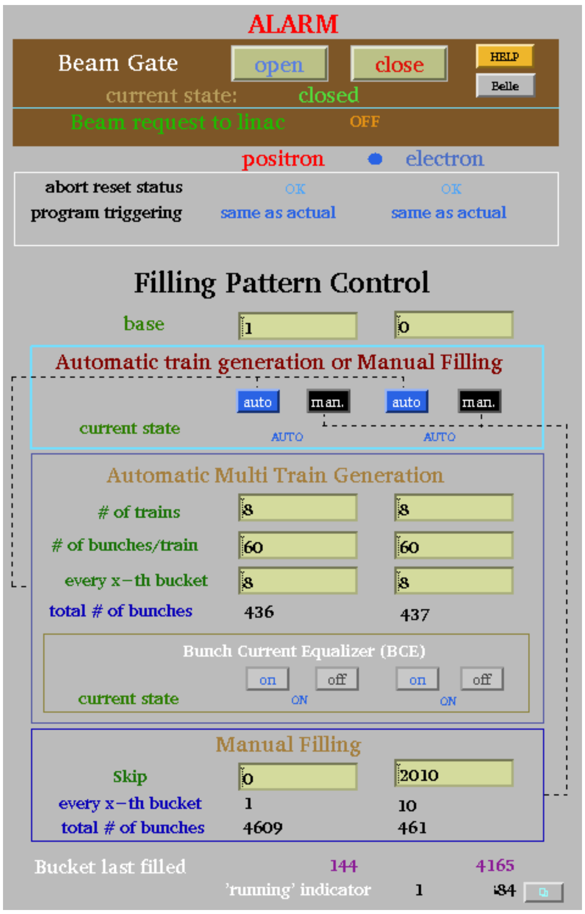


Figure 3: Control panel of the bucket selector system.

## 4.4 *Software on the linac computer*

A flow-chart of the program in the linac computer is shown in Fig. 4. This is coded not as device support but as a simple C-language program under VxWorks. The hardware accessed by the code are the Shared Memory Board and the delay modules, both of which are VME modules.

The code consists of some flow-control part and the calculation/setting of the delays including the 12 access times of the Shared Memory and 5 access times of the delay modules. Execution time for one trigger was measured to be about 40 micro seconds with the bunch-current equalizer off. A separate measurement showed us that one access of a VME module takes approximately 1 micro seconds. If the bunch-current equalizer is switched on, reading of the bunch-current information (4.6kbytes at most, 5ms) will be added. Adding the processing time of the interrupt under VxWorks, 21 micro seconds, the total time should be some 5ms, which is much shorter than 20ms.

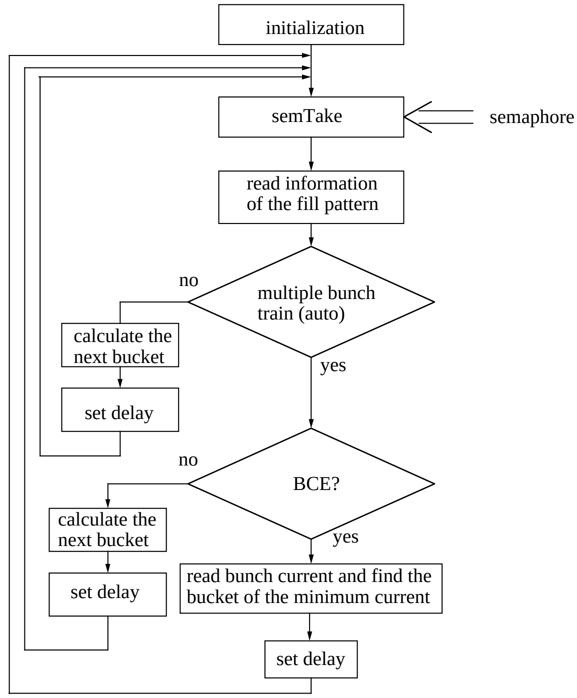


Figure 4: Flow chart of the code in the linac computer. The execution of the program in the linac computer is initiated by an interrupt from the bunch-current computer. The interrupt management is done with code prepared under the VxWorks System.

# 5 EXPERIENCE

With the bucket selector system, the beams are smoothly injected to the rings. When the operation scheme is changed, changing of the optics, for example, the optimum filling patterns are easily surveyed with this system.

# 6 REFERENCES


[1] K. Oide *et al.*, in Proceedings of the 1999 Particle Accelerator Conference, p.288.

[2] A. Enomoto, in Proceedings of the 1-st Asian Particle Accelerator Conference (KEK Proceedings 98-10), p.292.

[3] E. Kikutani *et al.*, in Proceedings of the 1-st Asian Particle Accelerator Conference (KEK Proceedings 98-10), p.525.

[4] M. Tobiyama *et al.*,, in preparation.

[5] A. Akiyama *et al.*, in Proceedings of the 1999 Particle Accelerator Conference, p.343.