

# DSCR (Data Stream Control Register)

DSCR register in powerpc allows user to have some control of prefetch of data stream in the processor. Please refer to the ISA documents or related manual for more detailed information regarding how to use this DSCR to attain this control of the prefetches. This document here provides an overview of kernel support for DSCR, related kernel objects, its functionalities and exported user interface.

## A. Data Structures:

### 1. thread\_struct:

```
dscr          /* Thread DSCR value */
dscr_inherit   /* Thread has changed default DSCR */
```

### 2. PACA:

```
dscr_default   /* per-CPU DSCR default value */
```

### 3. sysfs.c:

```
dscr_default   /* System DSCR default value */
```

## B. Scheduler Changes:

Scheduler will write the per-CPU DSCR default which is stored in the CPU's PACA value into the register if the thread has `dscr_inherit` value cleared which means that it has not changed the default DSCR till now. If the `dscr_inherit` value is set which means that it has changed the default DSCR value, scheduler will write the changed value which will now be contained in thread struct's `dscr` into the register instead of the per-CPU default PACA based DSCR value.

NOTE: Please note here that the system wide global DSCR value never gets used directly in the scheduler process context switch at all.

## C. SYSFS Interface:

- Global DSCR default: `/sys/devices/system/cpu/dscr_default`
- CPU specific DSCR default: `/sys/devices/system/cpu/cpuN/dscr`

Changing the global DSCR default in the sysfs will change all the CPU specific DSCR defaults immediately in their PACA structures. Again if the current process has the `dscr_inherit` clear, it also writes the new value into every CPU's DSCR register right away and updates the current thread's DSCR value as well.

Changing the CPU specific DSCR default value in the sysfs does exactly the same thing as above but unlike the global one above, it just changes stuff for that particular CPU instead for all the CPUs on the system.

## D. User Space Instructions:

The DSCR register can be accessed in the user space using any of these two SPR numbers available for that purpose.

1. Problem state SPR: 0x03 (Un-privileged, POWER8 only)
2. Privileged state SPR: 0x11 (Privileged)

Accessing DSCR through privileged SPR number (0x11) from user space works, as it is emulated following an illegal instruction exception inside the kernel. Both `mf spr` and `mt spr` instructions are emulated.

Accessing DSCR through user level SPR (0x03) from user space will first create a facility unavailable exception. Inside this exception handler all `mf spr` instruction based read attempts will get emulated and returned where as the first `mt spr` instruction based write attempts will enable the DSCR facility for the next time around (both for read and write) by setting DSCR facility in the FSCR register.

## E. Specifics about 'dscr\_inherit':

The thread struct element '`dscr_inherit`' represents whether the thread in question has attempted and changed the DSCR itself using any of the following methods. This element signifies whether the thread wants to use the CPU default DSCR value or its own changed DSCR value in the kernel.

1. `mt spr` instruction (SPR number 0x03)
2. `mt spr` instruction (SPR number 0x11)
3. `ptrace` interface (Explicitly set user DSCR value)

Any child of the process created after this event in the process inherits this same behaviour as well.

