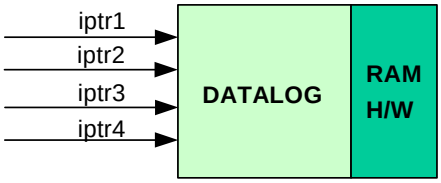
DATALOG	*4-Channel Data Logging Utility Module*

Description

This module stores the real-time values of four user selectable software Q15 variables in the data RAM provided on the 28xx DSP. Four variables are selected by configuring four module inputs, *iptr1, iptr2, iptr3* and *iptr4*, point to the address of the four variables. The starting addresses of the four RAM locations, where the data values are stored, are set to DLOG_4CH_buff1, DLOG_4CH_buff2, DLOG_4CH_buff3, and DLOG_4CH_buff4. The DATALOG buffer size, prescalar, and trigger value are also configurable.



Availability

This 16-bit module is available in one interface format:

1) The CcA interface version

Module Properties

Type: Target Dependent, Application Independent

Target Devices: Fixed Point x280x devices orPiccolo

CcA Version File Names: dlog4chc.asm, dlog4ch.h

IQmath library files for C: N/A

C Interface

Object Definition

The structure of DLOG_4CH object is defined by following structure definition

```
typedef struct {    long  task;         // Variable:  Task address pointer
                    int  *iptr1;        // Input: First input pointer (Q15)
                    int  *iptr2;        // Input: Second input pointer (Q15)
                    int  *iptr3;        // Input: Third input pointer (Q15)
                    int  *iptr4;        // Input: Fourth input pointer (Q15)
                    int  trig_value;    // Input: Trigger point (Q15)
                    int  prescalar;     // Parameter: Data log prescale
                    int  skip_cntr;     // Variable:  Data log skip counter
                    int  cntr;          // Variable:  Data log counter
                    long write_ptr;     // Variable:  Graph address pointer
                    int  size;          // Parameter: Maximum data buffer
                    int  (*init)();     // Pointer to init function
                    int  (*update)();   // Pointer to update function
                } DLOG_4CH;

typedef DLOG_4CH *DLOG_4CH_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	iptr1	Input pointer for the first Q15 variable	Q0	00000000-FFFFFFFF
	iptr2	Input pointer for the second Q15 variable	Q0	00000000-FFFFFFFF
	iptr3	Input pointer for the third Q15 variable	Q0	00000000-FFFFFFFF
	iptr4	Input pointer for the fourth Q15 variable	Q0	00000000-FFFFFFFF
Outputs	N/A	Data RAM	N/A	N/A
DATALOG Parameter	prescalar	Data log prescaler	Q0	0000-7FFF
	trig_value	Trigger point based on the fist Q15 variable	Q15	8000-7FFF
	size	Maximum data buffer	Q0	0000-7FFF
Internal	skip_cntr	Data log skip counter	Q0	0000-7FFF
	cntr	Data log counter	Q0	0000-7FFF
	write_ptr	Graph address pointer	Q0	00000000-FFFFFFFF
	task	Task address pointer	Q0	00000000-FFFFFFFF

Note: The trigger value is with reference to the input *iptr1. In accordance with this, the input connected to channel 1 should be time varying, and the trigger value should be set up such that input crosses the trigger value.

The other channels are captured synchronous to the channel 1. There is no trigger mechanism on channels 2 through 4.

Special Constants and Data types

DLOG_4CH

The module definition is created as a data type. This makes it convenient to instance an interface to the DATALOG driver. To create multiple instances of the module simply declare variables of type DLOG_4CH.

DLOG_4CH_handle
User defined Data type of pointer to DATALOG module

DLOG_4CH_DEFAULTS
Structure symbolic constant to initialize DATALOG module. This provides the initial values to the terminal variables as well as method pointers.

Methods

int DLOG_4CH_init(DLOG_4CH *);
int DLOG_4CH_update(DLOG_4CH *);

This default definition of the object implements two methods – the initialization and the runtime update function for DATALOG. This is implemented by means of a function pointer, and the initializer sets this to DLOG_4CH_init and DLOG_4CH_update functions for x281x/x280x. The argument to this function is the address of the DATALOG object.

Module Usage

Instantiation
The following example instances one DATALOG object
DLOG_4CH dlog1;

Initialization
To Instance pre-initialized objects
DLOG_4CH dlog1 = DLOG_4CH_DEFAULTS;

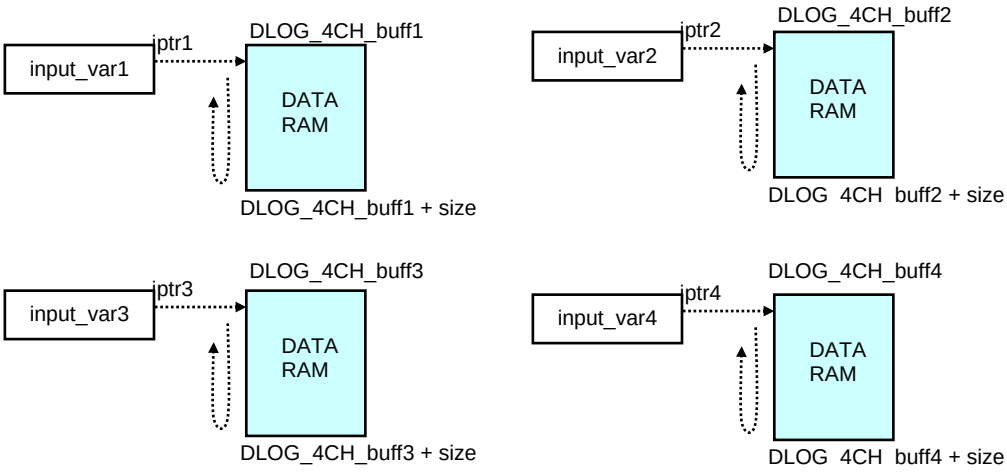
Invoking the computation function
dlog1.init(&dlog1);
dlog1.update(&dlog1);

Example
The following pseudo code provides the information about the module usage.

```
main()
{

        dlog1.iptr1 = &Q15_var1;          // Pass input to DATALOG module
        dlog1.iptr2 = &Q15_var2;          // Pass input to DATALOG module
        dlog1.iptr3 = &Q15_var3;          // Pass input to DATALOG module
        dlog1.iptr4 = &Q15_var4;          // Pass input to DATALOG module
        dlog1.trig_value = 0x0;           // Pass input to DATALOG module
        dlog1.size = 0x400;               // Pass input to DATALOG module
        dlog1.prescalar = 1;              // Pass input to DATALOG module
        dlog1.init(dlog1);                // Call init function for dlog1
}

void interrupt periodic_interrupt_isr()
{

        dlog1.update(&dlog1);             // Call update function for dlog1

}
```

Technical Background

This software module stores up to four real-time Q15 values of each of the selected input variables in the data RAM as illustrated in the following figures. The starting addresses of four RAM sections, where the data values are stored, are set to DLOG_4CH_buff1, DLOG_4CH_buff2, DLOG_4CH_buff3, and DLOG_4CH_buff4.



To show four stored Q15 variables in CCS graphs properly, the properties of two dual time graphs for these variables should be configured as shown in the following figures. In the software, the default buffer size is 0x400. The sampling rate is usually same as ISR frequency. In this case, it is 20 kHz with the prescalar of 1.