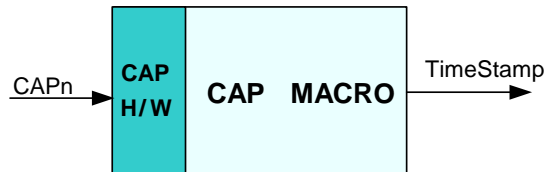


Description

This module provides the instantaneous value of the selected time base (GP Timer) captured on the occurrence of an event. Such events can be any specified transition of a signal applied at ECAP input pins of 280x devices.

**Availability**

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

- 1) The C interface version

Module Properties

Type: Target Dependent, Application Independent

Target Devices: 28x Fixed Point or Piccolo

C Version File Names: f2803xcap.h (for x2803x)

IQmath library files for C: N/A

C Interface

Object Definition

The structure of CAPTURE object is defined by following structure definition for

x280x series

```
typedef struct { int32 EventPeriod;    // Output: Timer value difference between two edges (Q0)
                } CAPTURE;
```

```
typedef CAPTURE *CAPTURE_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	CAPn (n=1,2,3,4)	Capture input signals to 28x device	N/A	0-3.3 v
Outputs	EventPeriod (x280x)	Timer value difference between two edges detected.	0	80000000-7FFFFFFF

Special Constants and Data types

CAPTURE

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

CAPTURE_handle

User defined Data type of pointer to CAPTURE module

CAPTURE_DEFAULTS

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

Methods

```
CAP_INIT_MACRO(CAPTURE *);
CAP_MACRO(CAPTURE *);
```

This default definition of the object implements two methods – the initialization and the runtime compute macro for CAPTURE generation. This is implemented by means of a macro pointer, and the initializer sets this to CAP_INIT_MACRO and CAP_MACRO macros for x280x. The argument to this macro is the address of the CAPTURE object.

Module Usage

Instantiation

The following example instances one CAPTURE object
CAPTURE cap1;

Initialization

To Instance pre-initialized objects
CAPTURE cap1 = CAPTURE_DEFAULTS;

Invoking the computation macro

CAP_INIT_MACRO (cap1);
CAP_MACRO (cap1);

Example

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

```
main()
{
    CAP_INIT_MACRO(cap1);                // Call init macro for cap1
}

void interrupt periodic_interrupt_isr()
{
    Uint16 Status;
    Uint32 EventPeriod;

    status = CAP_MACRO(cap1);            // Call the capture read macro

    // if status==1 then a time stamp was not read,
    // if status==0 then a time stamp was read

    if(status==0)
    {
        EventPeriod=(int32)(cap1.EventPeriod);    // Read out new time stamp
    }
}
```