



Enabling Time Synchronization (IEEE1588) Hardware on Intel® IXP43X Product Line of Network Processors

Application Note

October 2007



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice. The Intel® IXP4XX Product Line of Network Processors may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This Application Note as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Intel, Intel logo, and Intel XScale are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2007-2008, Intel Corporation



Contents

1.0	Introduction	5
1.1	Purpose	5
1.2	Intended Audience	5
1.3	Acronyms.....	5
1.4	Related Documents	5
2.0	Brief Introduction of Time Sync Access Layer	6
2.1	Enabling TSYNC via Time Sync Access Layer	6
2.2	Modifications in the Intel® IXP400 Software.....	6
2.2.1	Modification on Makefile	6
2.2.2	Modification on Time Sync Access Layer Component	6
2.2.3	Modification on Time Sync Codelet.....	7



Revision History

Date	Revision	Description
October 2007	001	Initial release

§ §



1.0 Introduction

The IEEE* 1588 standard defines a precision clock synchronization protocol for Networked Measurement and Control Systems. It provides a kind of correction mechanism by implementing several message exchange timing information to synchronize the individual clocks of the multiple systems.

The Intel® IXP43X Product Line of Network Processors provides hardware assist block to achieve this purpose. The hardware assist block is called “Time Synchronization Hardware Assist” (TSYNC). Refer to the Intel® IXP43X Product Line of Network Processors Developer’s Manual for detailed information.

1.1 Purpose

The intention of this application note is to guide you on the changes needed in the Intel® IXP400 Software to enable Time Synchronization Hardware Assist block on the IXP43X product line. In IXP400 software, this access layer module is called “Time Sync Access Layer” or “timeSyncAcc”.

The IXP400 software Time Sync Access Layer is already being used to enable TSYNC block on the Intel® IXP46X Product Line; this application note will guide you on how to enable the TSYNC block for the Intel® IXP43X Product Line of Network Processors.

Note: This document is targeted for Intel® IXP400 Software v2.4 and above only as the Intel® IXP400 Software v2.4 and above alone support Intel® IXP43X Product Line of Network Processors.

1.2 Intended Audience

The intended audience for this application note are those who have fundamental knowledge of Intel® IXP400 Software, IXP43X product line, and would like to enable the TSYNC hardware block on the IXP43X product line.

1.3 Acronyms

Acronym	Description
TSYNC	Time Synchronization Hardware Assist

1.4 Related Documents

Document Title	Document Number
<i>Intel® IXP43X Product Line of Network Processors Developer's Manual</i>	316843
<i>Intel® IXP400 Software Programmer's Guide v2.4</i>	252539
<i>Intel® IXP400 Software Programmer's Guide v3.0</i>	252539



2.0 Brief Introduction of Time Sync Access Layer

The Time Sync Access Layer that is used to enable the TSYNC comprises the following two software components:

Software component	Folder
TSYNC access layer software	/ixp400_xscale_sw/src/timeSyncAcc
TSYNC software codelet (sample application code)	/ixp400_xscale_sw/src/codelets/timeSyncAcc

Note: Software codelet is just a sample application code; the implementation may not achieve your requirement, and is not meant for production. You are required to rewrite your own application software.

2.1 Enabling TSYNC via Time Sync Access Layer

To enable the IXP43X product line via IXP400 software, you should modify the following software areas:

- Makefile
- timeSyncAcc
- timeSyncAcc codelet

2.2 Modifications in the Intel® IXP400 Software

You need to modify the following on the Intel® IXP400 Software v2.4 and above to enable TSYNC.

Note: The **Blue text** below denotes the additional code that must be added.

2.2.1 Modification on Makefile

Modification of Makefile is required to enable build of timeSyncAcc access layer component and codelet:

1. File: **/ixp400_xscale_sw/Makefile_ixp43X**

Modification:

- a. Locate label "BI_ENDIAN_COMPONENTS" and change accordingly as below:

```
BI_ENDIAN_COMPONENTS := atmdAcc atmM atmsch timeSyncAcc npeMh npeDI .....
```

- b. Locate label "BI_ENDIAN_CODELETS_COMPONENTS" and change accordingly as below:

```
BI_ENDIAN_CODELETS_COMPONENTS := hssAcc ethAcc atm timeSyncAcc
```

2.2.2 Modification on Time Sync Access Layer Component

Modification of the access layer is required to enable Time Sync Access Layer for the IXP43X product line:



1. File: **/ixp400_xscale_sw/src/include/IxTimeSyncAcc.h**

Modification:

- a. Locate label "#if defined (__ixp46X)" in IxTimeSyncAcc.h, and change accordingly as below:

```
#if defined (__ixp46X) || defined (__ixp43X)
```

2. File: **/ixp400_xscale_sw/src/timeSyncAcc/IxTimeSyncAcc.c**

Modification:

- a. Locate label "#if defined (__ixp46X)" in IxTimeSyncAcc.c, and change accordingly as below:

```
#if defined (__ixp46X) || defined (__ixp43X)
```

- b. Locate function *ixTimeSyncAccInitCheck*, add read device API of IXP43X product line as shown below:

```
/* Check for IXP46X and IXP43X device */
if((IX_FEATURE_CTRL_DEVICE_TYPE_IXP46X != ixFeatureCtrlDeviceRead()) &&
    (IX_FEATURE_CTRL_DEVICE_TYPE_IXP43X != ixFeatureCtrlDeviceRead()))
{
```

3. File: **/ixp400_xscale_sw/src/timeSyncAccSymbols.c**

- a. Locate label "#if defined (__ixp46X)" in IxTimeSyncAccSymbols.c, change it as shown below:

```
#if defined (__ixp46X) || defined (__ixp43X)
```

2.2.3 Modification on Time Sync Codelet

To enable timeSyncAcc codelet for the IXP43X product line, you need to make many changes in the timeSyncAcc codelet. Follow the instructions below:

1. File: **/ixp400_xscale_sw/src/codelets/timeSyncAcc/IxTimeSyncAccCodelet.c**

Modification:

- a. Locate the label "#if defined (__ixp46X)" in IxTimeSyncAccCodelet.c, change it as shown below:

```
#if defined (__ixp46X) || defined (__ixp43X)
```



- b. Locate the function - *ixTimeSyncAccCodeletMain* and make the following changes to this function:

```
/* configure all Time Sync channels */
for (tsChannel = IX_TIMESYNACC_NPE_A_1588PTP_PORT;
tsChannel < IX_TIMESYNACC_CODELET_MAX_TS_CHANNELS;
tsChannel++)
{
#ifdef __ixp43X
if(tsChannel == IX_TIMESYNACC_NPE_B_1588PTP_PORT)
{
continue;
}
#endif

/* get channel operating mode from the configuration table */
tsChannelMode = ixTimeSyncAccCodeletConfigPtr->tsChannelMode[tsChannel];
```

- c. Locate the function - *ixTimeSyncAccCodeletEthInit* and make the following changes to this function:

```
/* check if the device is IXP46X or IXP43X */
if ((IX_FEATURE_CTRL_DEVICE_TYPE_IXP46X != ixFeatureCtrlDeviceRead ()) &&
(IX_FEATURE_CTRL_DEVICE_TYPE_IXP43X != ixFeatureCtrlDeviceRead ()))
{
ixOsallLog (IX_OSAL_LOG_LVL_ERROR, IX_OSAL_LOG_DEV_STDERR,
"ixTimeSyncAccCodeletEthInit: this device is not IXP46X\n",
0, 0, 0, 0, 0, 0);
return IX_FAIL;
}

/* check if all NPEs are enabled */
for (channel = 0, npe = IX_FEATURECTRL_NPEA; npe <= IX_FEATURECTRL_NPEC; npe++,
channel++)
{
#ifdef __ixp43X
if(npe == IX_FEATURECTRL_NPEB)
{
continue;
}
#endif
if (IX_FEATURE_CTRL_COMPONENT_ENABLED != ixFeatureCtrlComponentCheck (npe))
```

```
ixTimeSyncAccCodeletUninitFuncMap[IX_TIMESYNACC_CODELET_NPE_A].initialized = TRUE;
ixTimeSyncAccCodeletUninitFuncMap[IX_TIMESYNACC_CODELET_NPE_DL].initialized = TRUE;

#ifdef __ixp46X
/* download NPE B's image with basic Ethernet Rx/Tx and activate NPE B */
if (IX_SUCCESS != ixNpeDINpeInitAndStart (IX_NPEDL_NPEIMAGE_NPEB_ETH))
{
ixOsallLog (IX_OSAL_LOG_LVL_ERROR, IX_OSAL_LOG_DEV_STDERR,
"ixTimeSyncAccCodeletEthInit: failed to initialize and start NPE B\n",
0, 0, 0, 0, 0, 0);
return IX_FAIL;
}
ixTimeSyncAccCodeletUninitFuncMap[IX_TIMESYNACC_CODELET_NPE_B].initialized = TRUE;
#endif

/* download NPE C's image with basic Ethernet Rx/Tx and activate NPE C */
if (IX_SUCCESS != ixNpeDINpeInitAndStart (IX_NPEDL_NPEIMAGE_NPEC_ETH))
```



```

/* initialize all ethernet ports */
for (portId = IX_ETH_PORT_1; portId <= IX_ETH_PORT_3; portId++)
{
#ifdef __ixp43X
    if(portId == IX_ETH_PORT_1)
    {
        continue;
    }
#endif
    if (IX_ETH_ACC_SUCCESS != ixEthAccPortInit (portId))

```

- d. Locate the function - *ixTimeSyncAccCodeletPortDisable* and make the following changes to this function:

```

for (portId = IX_ETH_PORT_1; portId <= IX_ETH_PORT_3; portId++)
{
#ifdef __ixp43X
    if(portId == IX_ETH_PORT_1)
    {
        continue;
    }
#endif

    status = ixEthAccPortDisable (portId);

```

- e. Locate the function - *ixTimeSyncAccCodeletMbufsFree* and make the following changes to this function:

```

/* free mBufs */
for (count = 0; count < IX_TIMESYNACC_CODELET_MAX_TS_CHANNELS; count++)
{
#ifdef __ixp43X
    if(count == IX_TIMESYNACC_NPE_B_1588PTP_PORT)
    {
        continue;
    }
#endif

    mBufPtr = ixTimeSyncAccCodeletGlobalMBuf[count];

```

- f. Locate the function - *ixTimeSyncAccCodeletPTPMsgTransmit* and make the following changes to this function:

```

for (tsChannel = IX_TIMESYNACC_NPE_A_1588PTP_PORT;
tsChannel < IX_TIMESYNACC_CODELET_MAX_TS_CHANNELS;
tsChannel++)
{
#ifdef __ixp43X
    if(tsChannel == IX_TIMESYNACC_NPE_B_1588PTP_PORT)
    {
        continue;
    }
#endif

    portId = ixTimeSyncAccCodeletPortIdList[tsChannel];

```



```
/* sleep and wait for interval time to elapse before transmitting next PTP message */
ixOsalSleep (IX_TIMESYNACC_CODELET_PTP_MSG_XMIT_INTERVAL);

for (portId = IX_ETH_PORT_1; portId <= IX_ETH_PORT_3; portId++)
{
#ifdef __ixp43X
    if(portId == IX_ETH_PORT_1)
    {
        continue;
    }
#endif

    if (IX_SUCCESS != ixOsalSemaphoreWait (&ixTimeSyncAccCodeletSemId,
IX_TIMESYNACC_CODELET_PTP_MSG_XMIT_INTERVAL))
    {
```

- g. Locate the function - *ixTimeSyncAccCodeletPTPMsgCheck* and make the following changes to this function:

```
for (channel = IX_TIMESYNACC_NPE_A_1588PTP_PORT;
channel < IX_TIMESYNACC_CODELET_MAX_TS_CHANNELS;
channel++)
{
#ifdef __ixp43X
    if(channel == IX_TIMESYNACC_NPE_B_1588PTP_PORT)
    {
        continue;
    }
#endif

    /* initialize ptpMsgData buffer */
    ixOsalMemSet ((void *)&ptpMsgData, 0xff, sizeof (IxTimeSyncAccPtpMsgData));
```

- 2. File: `\ixp400_xscale_sw\src\codelets\timeSyncAcc\IxTimeSyncAccCodelet.h`

- h. Locate the label “#if defined (__ixp46X)” in `IxTimeSyncAccCodelet.h` and change accordingly as shown below:

```
#if defined (__ixp46X) || defined (__ixp43X)
```

