

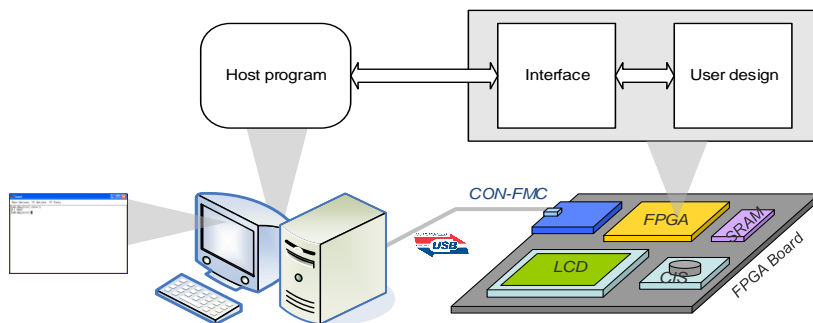
CON-FMC

Connect any computer to FPGA through USB

June 2019

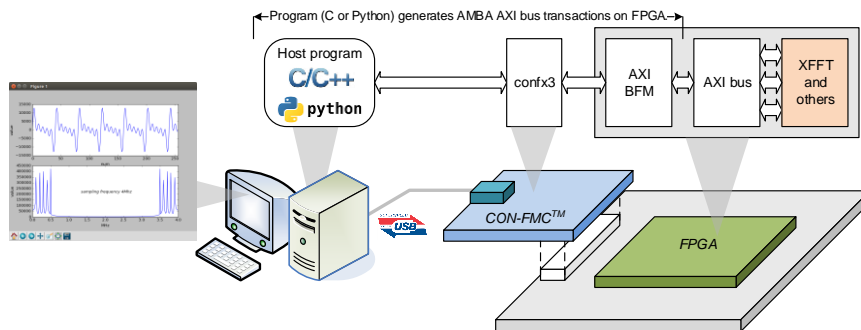
Future Design Systems
www.future-ds.com

Connect Linux/Windows/Android to your FPGA board through USB 3.0



An example

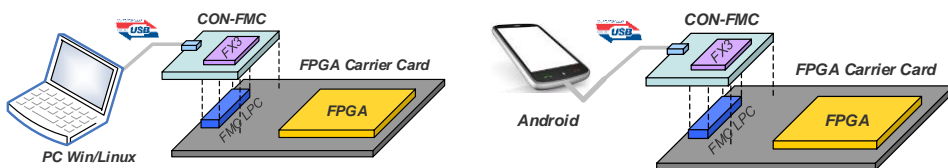
Running FFT HW core along with C/C++ or Python through USB 3.0



Copyright © 2017-2018-2019 Future Design Systems, Inc.

(3)

FUTURE
Design Systems



Connect any computer to FPGA through USB

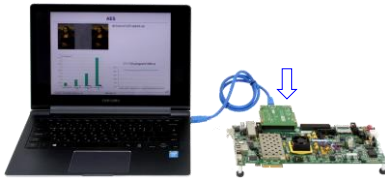
CON-FMC OVERVIEW

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(4)

FUTURE
Design Systems

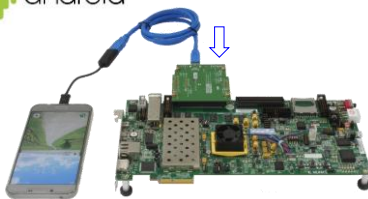
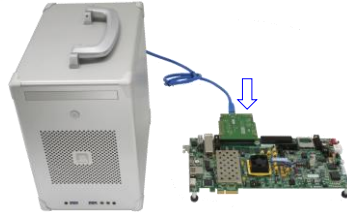
CON-FMC with OS



ubuntu

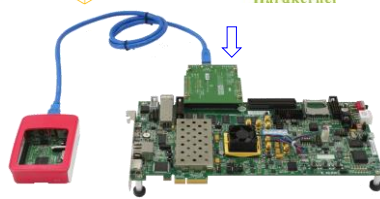


CentOS



ARTIK™

ODROID
Hardkernel



Copyright © 2017-2018-2019 Future Design Systems, Inc.

(5)

FUTURE
Design Systems

CON-FMC with FPGA board (carrier board)

✦ http://www.future-ds.com/en/products.html#CON_FMC

✦ ML605 (UCF-LPC, UCF-HPC)



✦ VC108 (XDC-HPC0, XDC-HPC1)



✦ ZCU111 (XDC-HSPC)



✦ ZC702 (XDC-LPC1, XDC-LPC2)



✦ ZedBoard (XDC-LPC)



✦ ZC706 (XDC-LPC, XDC-HPC)



✦ HTG-ZRF8 (XDC-HPC)



✦ FMC (FPGA Mezzanine Card)

✦ VITA

✦ <http://www.vita.com/>

✦ VITA FMC

✦ <http://www.vita.com/fmc>

✦ VITA 57.1.1

✦ HPC+/HPCP (HPC plus): 14x40 (560)

✦ HPC (High Pin Count): 10x40 (400)

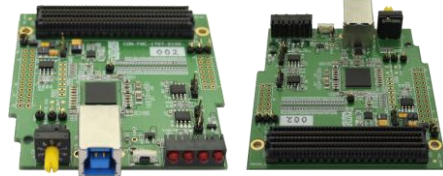
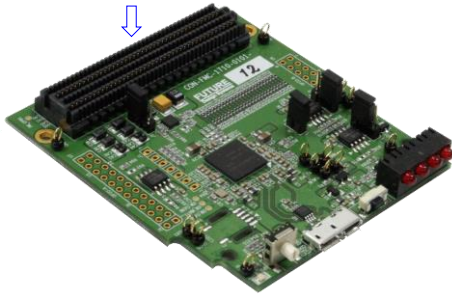
✦ LPC (Low Pin Count): 4x40 (160)

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(6)

FUTURE
Design Systems

CON-FMC



Companion products: CON-GPIO, CON-FPGA

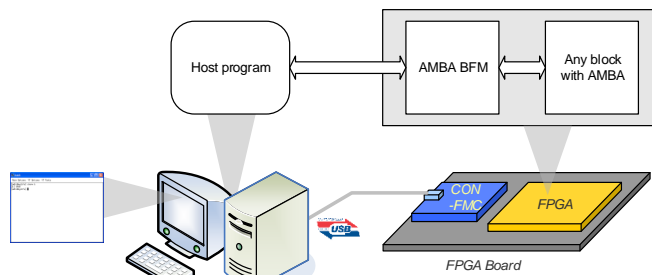


- USB 3.0 (5Gbit/sec) and 2.0 (480Mbit/sec)
 - ◆ Cypress FX3
- FMC/FMC+ (LPC, HPC, HPC+)
 - ◆ VITA 57.1 LPC
- Multi-board supporting up to 8
 - ◆ CID
- Platforms
 - ◆ Windows
 - ◆ Linux
 - ◆ Android (Java and native)
- Additional features
 - ◆ C API to develop user application
 - ◆ AMBA AXI/AHB/APB BFM
 - ◆ Python
- Performance
 - ◆ ~80~100Mbyte/sec (USB3.0)
 - ◆ ~10Mbyte/sec (USB2.0)

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(7)

FUTURE
Design Systems



Connect any computer to FPGA through USB

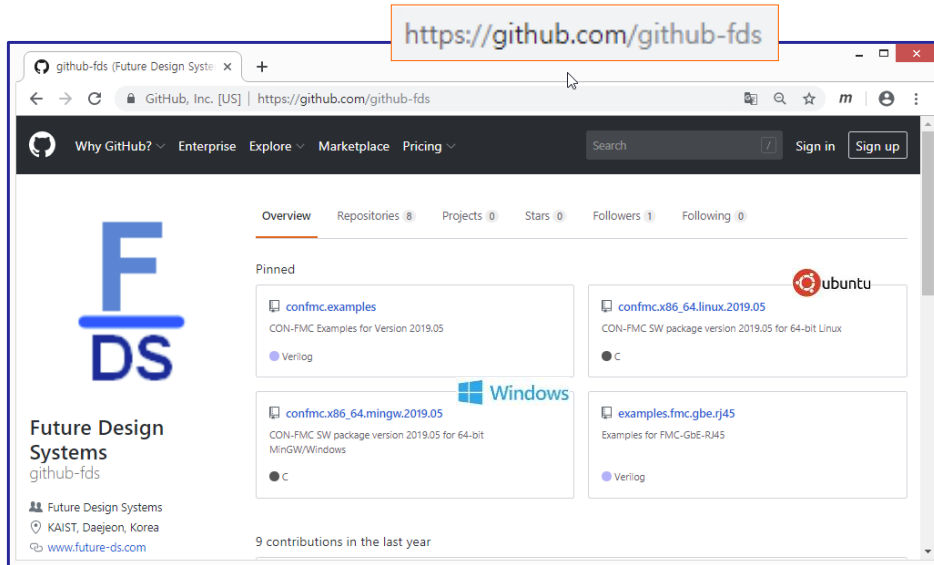
CON-FMC TYPICAL USAGES

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(8)

FUTURE
Design Systems

Easy to install from GitHub



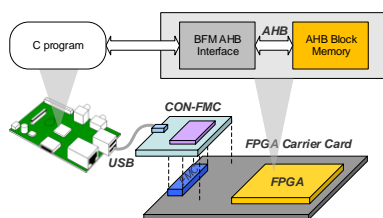
Copyright © 2017-2018-2019 Future Design Systems, Inc.

(9)

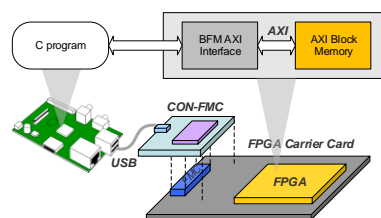
FUTURE
Design Systems

Example template for AMBA AXI and AHB

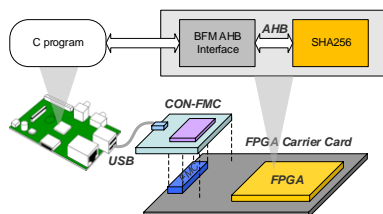
AMBA AHB memory



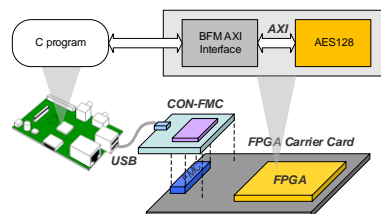
AMBA AXI memory



User case using AHB template for SHA256 (serial)



User case using AXI template for AES128 (pipelined)



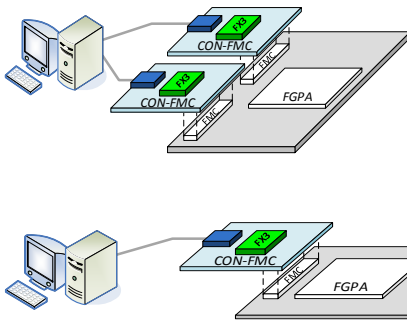
Copyright © 2018 Future Design Systems Inc.

(10)

CON-FMC multi-programming

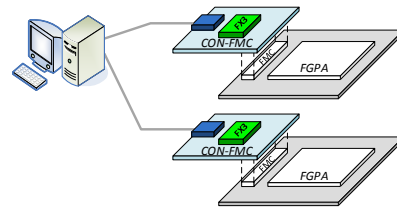
Multi-programming

- ◆ different programs access to different CID
 - Supported



Multi-threading

- ◆ the same program access different CID
 - Supported
 - (multi-threading requires user-level exclusion)
- ◆ different programs access the same CID
 - Not supported yet
 - (it requires system-wide exclusion)



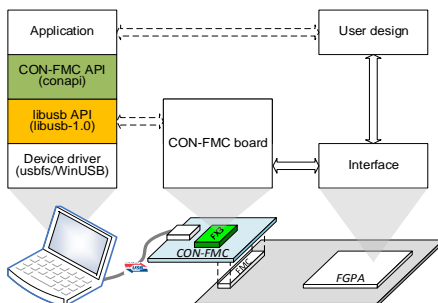
Copyright © 2017-2018-2019 Future Design Systems, Inc.

(11)

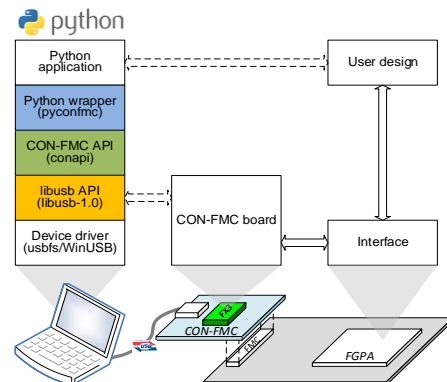
FUTURE
Design Systems

Programming interfaces

Standard C/C++ interface



Python interface



Copyright © 2017-2018-2019 Future Design Systems, Inc.

(12)

FUTURE
Design Systems

Simple API: C and C++

Minimum code: C

```
#include "conapi.h"

int main(int argc, char *argv[])
{
    unsigned int cid=0;
    unsigned int mode=CON_MODE_CMD;
    unsigned int loglevel=CONAPI_LOG_LEVEL_INFO;

    con_Handle_t handle=conInit(cid, mode, loglevel);
    conReset(handle, 1);
    cid=conGetCID(handle);
    conRelease(handle);

    return 0;
}
```

GNU GCC supporting
Visual Studio: 64-bit & 32-bit

API's

```
con_Handle_t conInit ( unsigned int con_cid
                      , unsigned int con_mode
                      , unsigned int log_level );
int conRelease ( con_Handle_t con_handle );
int conCmdWrite ( con_Handle_t con_handle
                , void *pBuffer
                , unsigned int nNumberOfItemsToWrite
                , unsigned int *pNumberOfItemsWritten
                , unsigned int transactor );
int conDataWrite( con_Handle_t con_handle
                , void *pBuffer
                , unsigned int nNumberOfItemsToWrite
                , unsigned int *pNumberOfItemsWritten
                , unsigned int transactor );
int conDataRead ( con_Handle_t con_handle
                , void *pBuffer
                , unsigned int nNumberOfItemsToRead
                , unsigned int *pNumberOfItemsRead
                , unsigned int transactor );
int conStreamWrite( con_Handle_t con_handle
                  , void *pBuffer
                  , unsigned int nNumOfItemsToWrite
                  , unsigned int *pNumOfItemsWritten
                  , unsigned int zlp );
int conStreamRead ( con_Handle_t con_handle
                  , void *pBuffer
                  , unsigned int nNumOfItemsToRead
                  , unsigned int *pNumOfItemsRead );
```

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(13)

FUTURE
Design Systems

Simple API wrappers

Bus Functional Module API

```
#ifdef __cplusplus
extern "C" {
#endif

void BfmWrite( con_Handle_t handle
              , unsigned int addr
              , unsigned int *data
              , unsigned int size
              , unsigned int length);
void BfmRead ( con_Handle_t handle
              , unsigned int addr
              , unsigned int *data
              , unsigned int size
              , unsigned int length);
void BfmWriteFix( con_Handle_t handle
                 , unsigned int addr
                 , unsigned int *data
                 , unsigned int size
                 , unsigned int length);
void BfmReadFix ( con_Handle_t handle
                 , unsigned int addr
                 , unsigned int *data
                 , unsigned int size
                 , unsigned int length);
int BfmGpout( con_Handle_t handle
             , unsigned int Value );
int BfmGpin ( con_Handle_t handle
             , unsigned int *pValue );
#ifdef __cplusplus
}
#endif
```

Python

```
import sys
import confmc

hdl = confmc.conInit()
if not hdl: sys.exit(1)

cid = confmc.conGetCid(hdl)
if cid<0: sys.exit(1)

print("CON-FMC:" + str(cid) + " found.")
confmc.conRelease(hdl)
```

Generates AXI or AHB write bus transaction with the given arguments:

- addr: address
- data: buffer containing data to be written
- size: num of bytes for each beat
- length: length of burst, i.e., num of beats

Copyright © 2017-2018-2019 Future Design Systems, Inc.

(14)

FUTURE
Design Systems

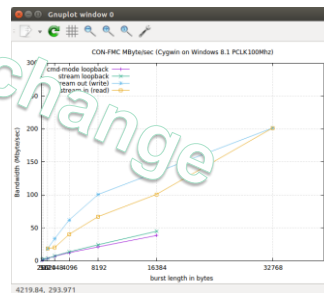
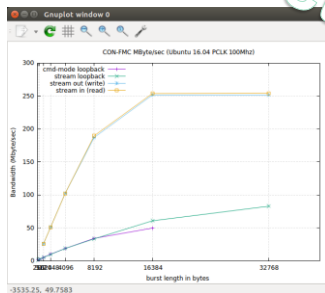
Bandwidth (highly depending on environments)

x86_64 Ubuntu 16.04 / USB 3.0

- ◆ ~50MB/sec for pseudo-DMA loopback (bandwidth for pure payload)
- ◆ ~85MB/sec for stream loopback
- ◆ ~250MB/sec for stream in or out

x86_64 Cygwin on Windows / USB 3.0

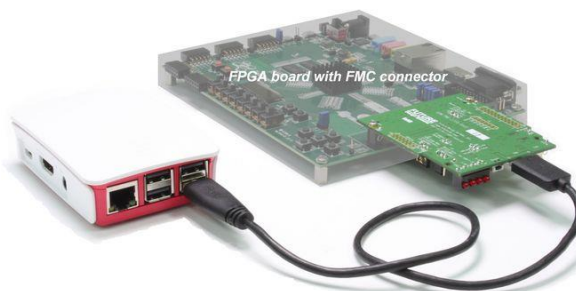
- ◆ ~45MB/sec for pseudo-DMA loopback
- ◆ ~45MB/sec for stream loopback
- ◆ ~90MB/sec for stream out
- ◆ ~120MB/sec for stream in



Copyright © 2017-2018-2019 Future Design Systems, Inc.

(15)

FUTURE
Design Systems



Connect any computer to FPGA through USB

CON-FMC EXAMPLES

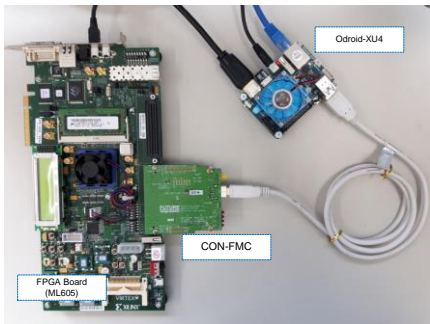
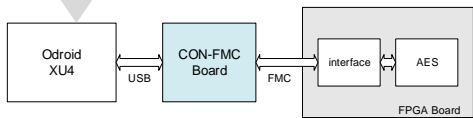
Copyright © 2017-2018-2019 Future Design Systems, Inc.

(16)

FUTURE
Design Systems

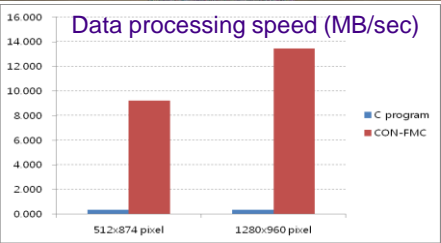
Example project

Image feeding & display program

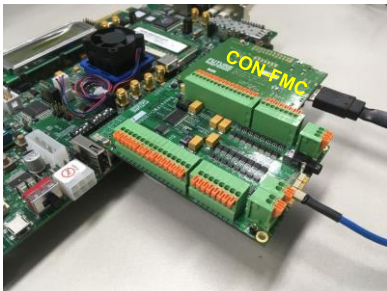


AES (Advance Encryption Standard)

- ~35 times fast result in terms of time comparing to pure software

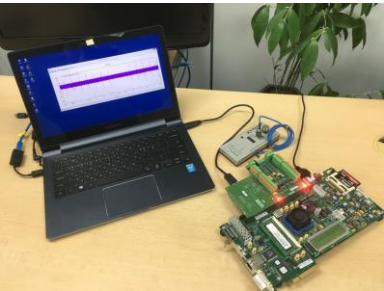
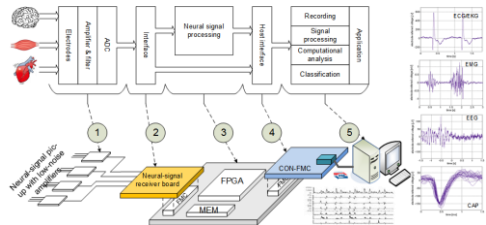


Example project

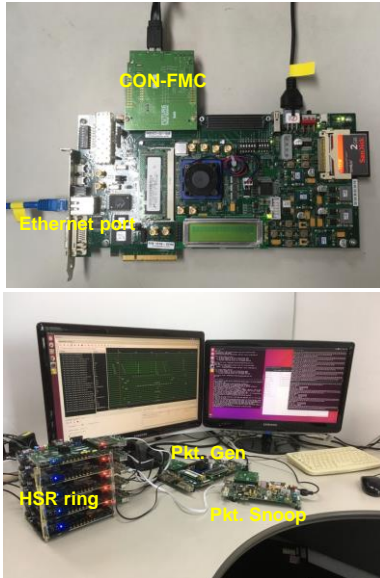


NeuralFMC system

- CON-FMC provides a path to receive neural signal data through USB.

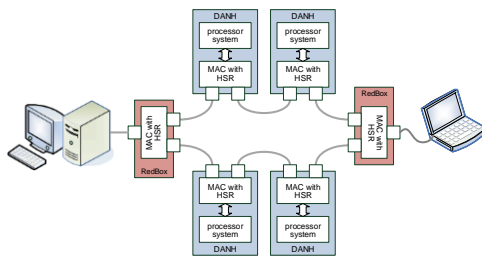


Example project



■ HSR/PTP (High-availability Seamless Redundancy and Precision Time Protocol)

- ◆ CON-FMC to inject user-defined Ethernet packets and to snoop packets on the line.



(주)퓨처디자인시스템

- 주소: 대전광역시 유성구 문지로 193,
KAIST 문지캠퍼스 학부동 F723호
- 전화: (042) 864-0211~2
- 전자우편: contact@future-ds.com
- 홈페이지: www.future-ds.com
- 위치: http://ouic.kaist.ac.kr/sub04_06

