

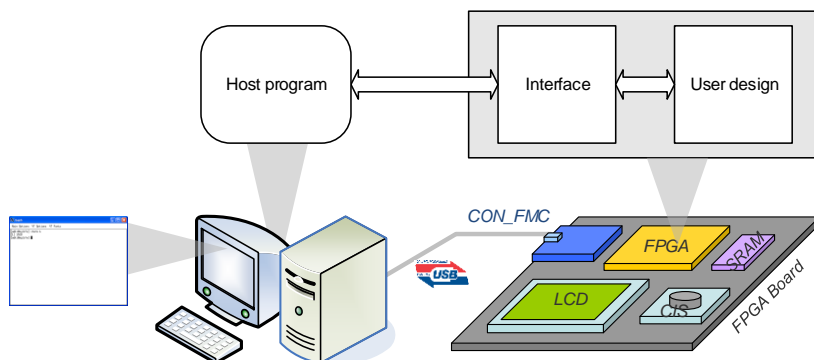
# CON-FMC

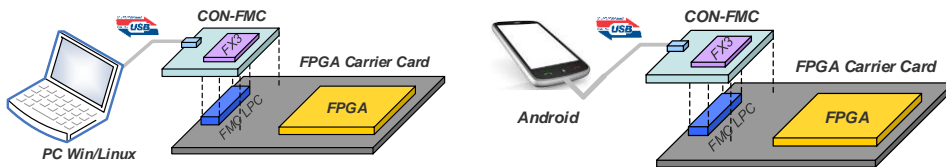
Connect any computer to FPGA through USB

March 2019

Future Design Systems  
[www.future-ds.com](http://www.future-ds.com)

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





Connect any computer to FPGA through USB

## CON-FMC OVERVIEW

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

( 3 )

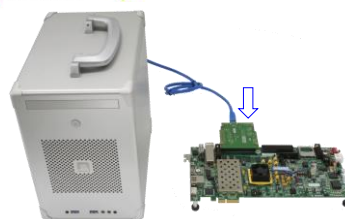
**FUTURE**  
Design Systems

## CON-FMC along with board

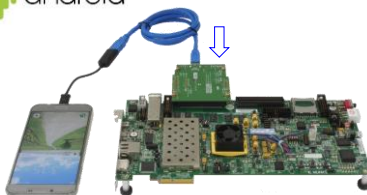
Windows



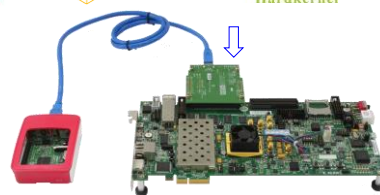
ubuntu CentOS



android



Raspberry Pi SAMSUNG ARTIK™ ODROID Hardkernel

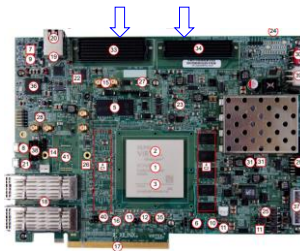
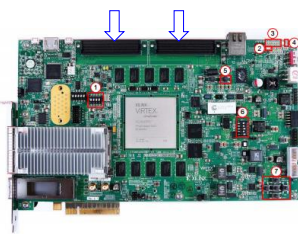
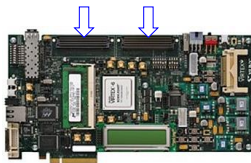


Copyright © 2017-2018-2019 Future Design Systems, Inc.

( 4 )

**FUTURE**  
Design Systems

## What is FMC



### FPGA Mezzanine Card

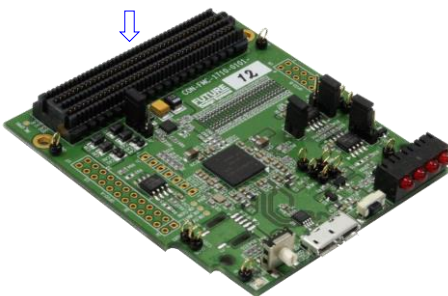
- ◆ VITA
  - ◆ <http://www.vita.com/>
- ◆ VITA FMC
  - ◆ <http://www.vita.com/fmc>
- ◆ VITA 57.1.1
  - ◆ HPC+/HPCP (HPC plus): 14x40 (560)
  - ◆ HPC (High Pin Count): 10x40 (400)
  - ◆ LPC (Low Pin Count): 4x40 (160)

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

( 5 )

**FUTURE**  
Design Systems

## CON-FMC



*Companion products: CON-GPIO, CON-FPGA*

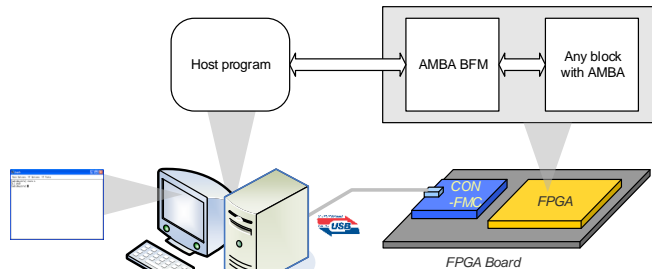


- ◆ USB 3.0 and 2.0
  - ◆ Cypress FX3
- ◆ FMC/FMC+ (LPC, HPC, HPC+)
  - ◆ VITA 57.1 LPC
- ◆ Multi-board supporting up to 8
  - ◆ CID
- ◆ Platforms
  - ◆ Windows
  - ◆ Linux
  - ◆ Android (Java and native)
- ◆ Additional features
  - ◆ C API to develop user application
  - ◆ AMBA AXI/AHB/APB BFM
- ◆ Performance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

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

( 6 )

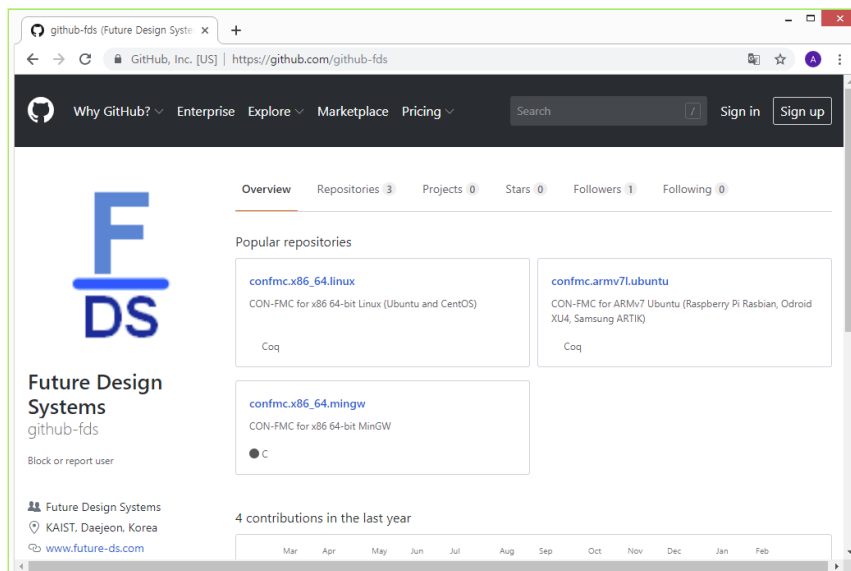
**FUTURE**  
Design Systems



Connect any computer to FPGA through USB

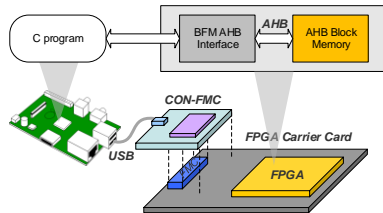
## CON-FMC TYPICAL USAGES

## Easy to install from GitHub

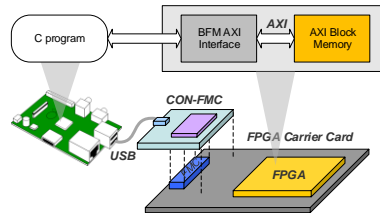


## Example template for AMBA AXI and AHB

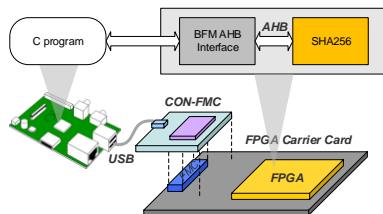
### AMBA AHB memory



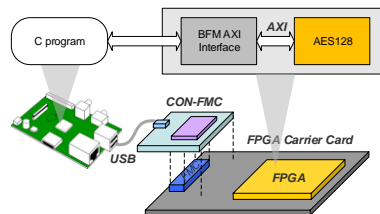
### AMBA AXI memory



### User case using AHB template for SHA256 (serial)



### User case using AXI template for AES128 (pipelined)



Copyright © 2018 Future Design Systems Inc.

(9)

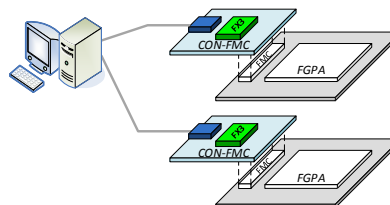
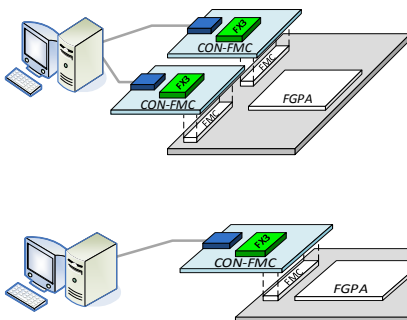
## CON-FMC multi-programming

### 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)



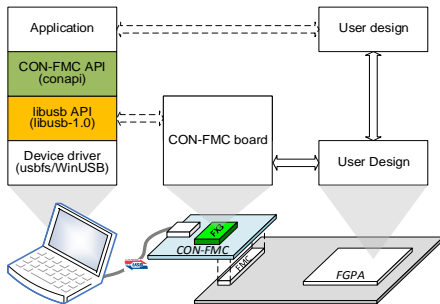
Copyright © 2017-2018-2019 Future Design Systems, Inc.

(10)

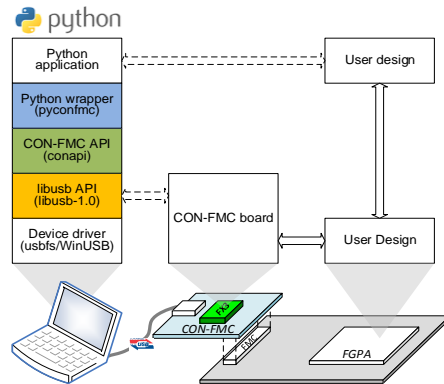
**FUTURE**  
Design Systems

## Programming interfaces

### Standard C/C++ interface



### Python interface



## 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;
}
```

### 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 );
```

GNU GCC supporting

## Simple API wrappers

### Bus Functional Module API

```
#ifndef __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

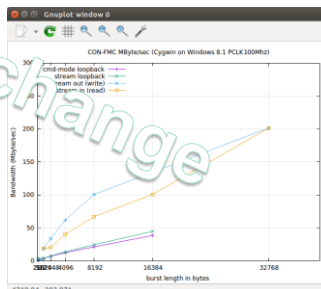
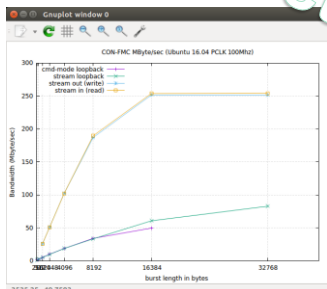
## Bandwidth (highly depending on environments)

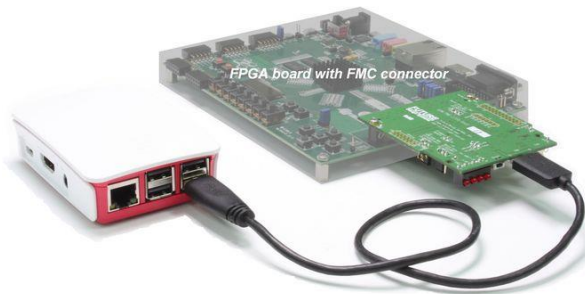
### x86\_64 Ubuntu 16.04 / USB 3.0

- ◆ ~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

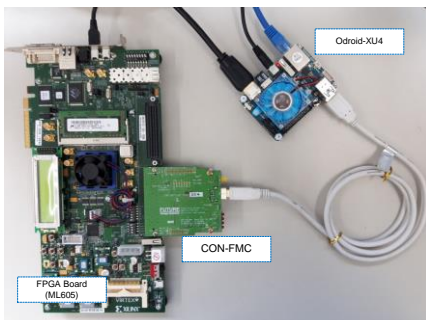
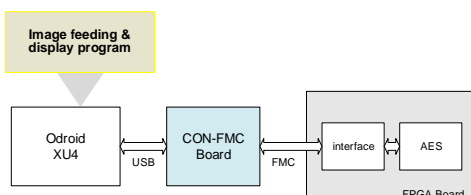




Connect any computer to FPGA through USB

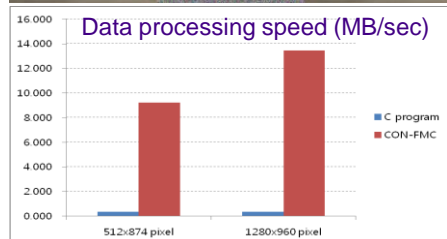
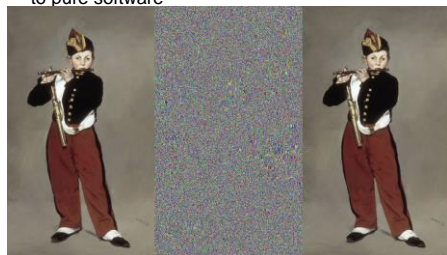
## CON-FMC EXAMPLES

## Example project



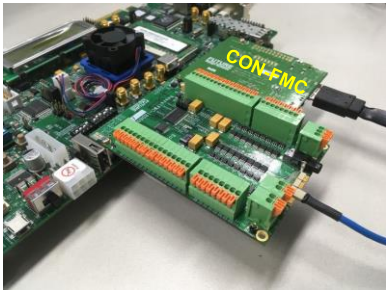
### ❖ AES (Advance Encryption Standard)

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

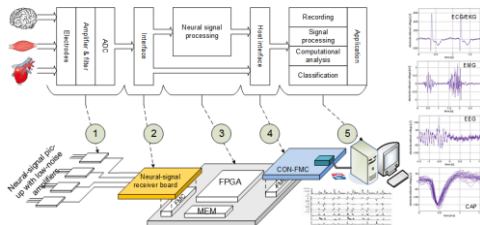




# Example project



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

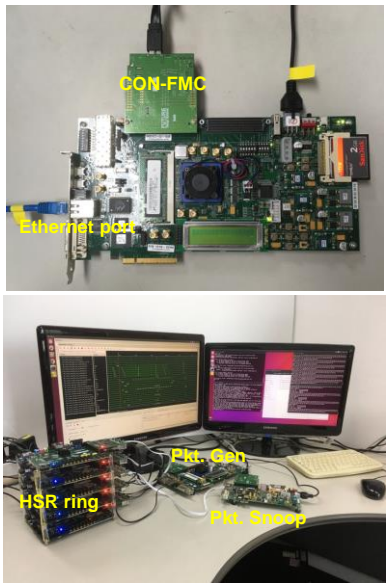


Copyright © 2017-2018-2019 Future Design Systems, Inc.

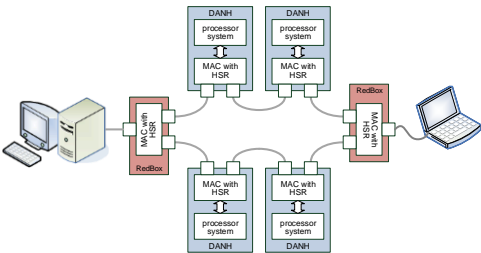
( 17 )

FUTURE  
Design Systems

# 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.



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

( 18 )

FUTURE  
Design Systems

## (주)퓨처디자인시스템

- 주소: 대전광역시 유성구 문지로 193,  
KAIST 문지캠퍼스 학부동 F723호
- 전화: (042) 864-0211~2
- 전자우편: [contact@future-ds.com](mailto:contact@future-ds.com)
- 홈페이지: [www.future-ds.com](http://www.future-ds.com)
- 위치: [http://ouic.kaist.ac.kr/sub04\\_06](http://ouic.kaist.ac.kr/sub04_06)



**FUTURE**  
Design Systems  
FUTURE  
Design Systems

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

( 19 )