Content:

1. Review on AMchip R&D
   - Activity time-line
   - AMchip features
   - Evolution of AMchip test board
   - Latest update of the test-board
2. Test-bench 1 (Virtex-6 FPGA)
   - Firmware/software
   - Architecture and key components
   - AMchip characterization and test-bench tuning
3. AMchip application in sequence matching
   - Smith-waterman algorithm for sequence matching
   - Reformulation of smith-waterman for AMchip
   - Implementation
   - results
4. IPMC JTAG interface
   - JTAG interface
5. Test-bench 2 (ZYNQ-7 FPGA)
   - Embedded system (hardware software integration)
   - Embedded Linux and interfacing with host processor
   - Embedded Linux and driving ZYNQ FPGA through AXI interface and DMA
   - 7 series high-speed GTX interface
6. Future plan and improvement
   - SATA IP development based on DMA
   - Integration of SATA in test-bench 2

AMchip:

Summary of activities, new research, development and applications

M. Ali Mirzaei

CNRS Postdoc, Micro-Electronics Engineer researcher

UPMC, LPNHE,

ATLAS group, AMchip team

Contact: mmirzaei@lpnhe.in2p3.fr

Mirzai142.nri@gmail.com

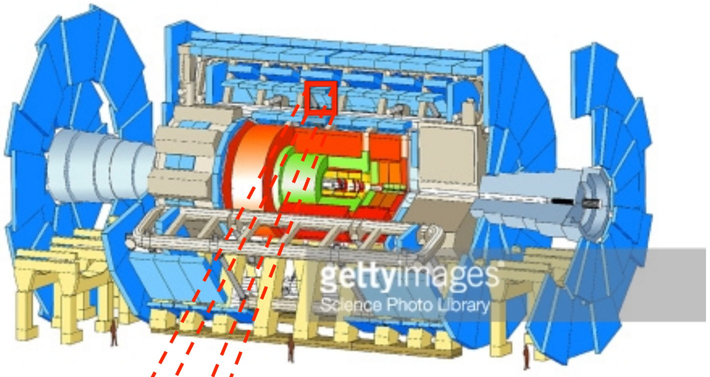
Test-bench 1(firmware/software)

Activities	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16	Mar-16	Apr-16	May-16	Jun-16	Jul-16	Aug-16	Sep-16	Oct-16	Nov-16	Dec-16	Jan-17	Feb-17	Mar-17
AMchip Test-bench 1 (firmware and software)	🟩	🟩	🟩	🟩	🟩	🟩	🟩																	
Sequence alignment based on AMchip																								
IPMC board																								
AMchip Test-bench 2 (firmware and software)																								
presentation and documentstation																								

AMchip in FTK and ATLAS detector

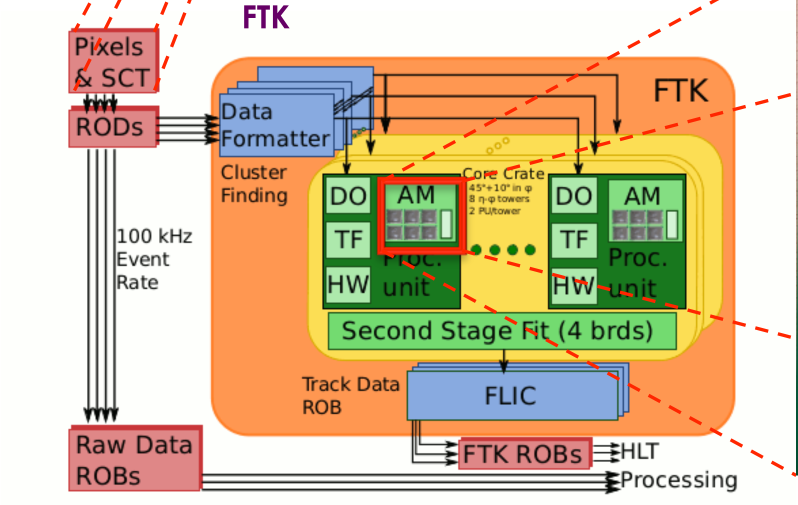
This presentation is all about "AMchip"

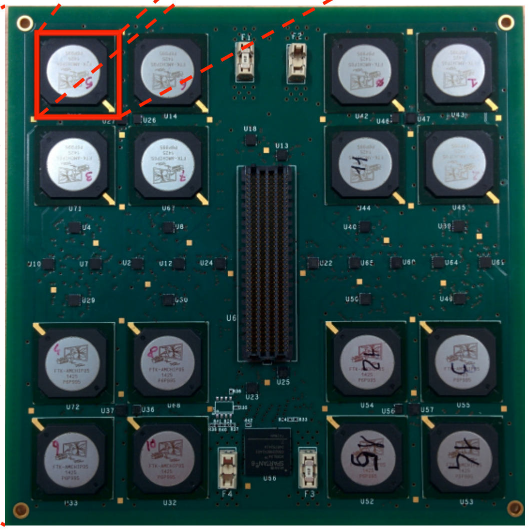
ATLAS detector



AMchip

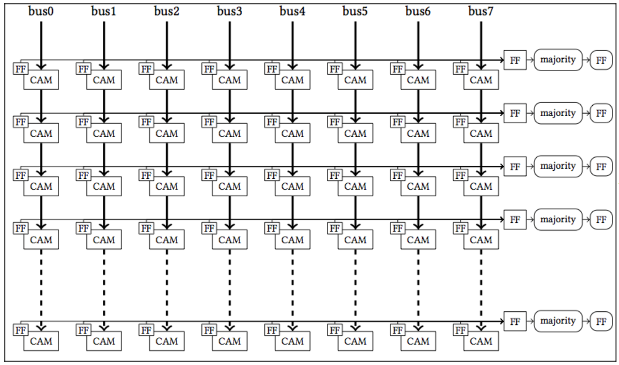
FTK

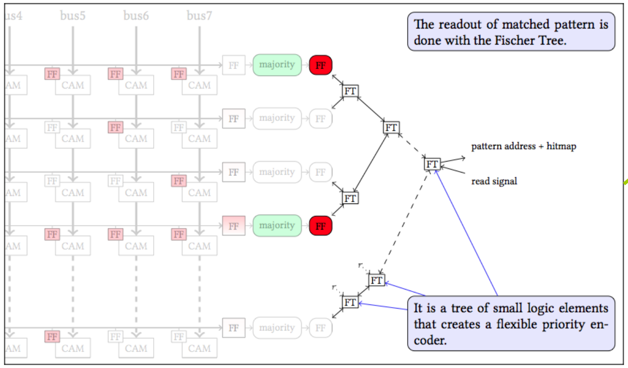




AM board

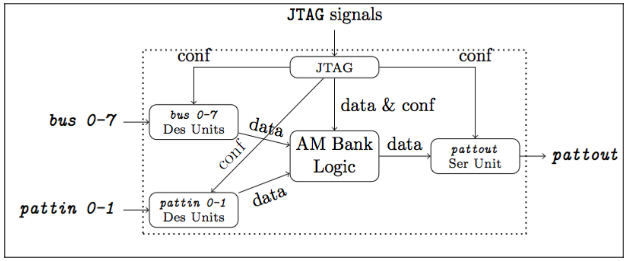
AMchip Architecture





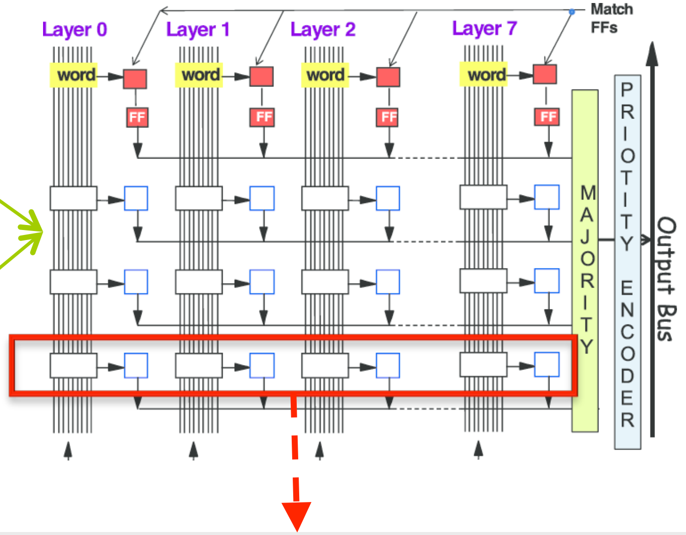
The readout of matched pattern is done with the Fischer Tree.

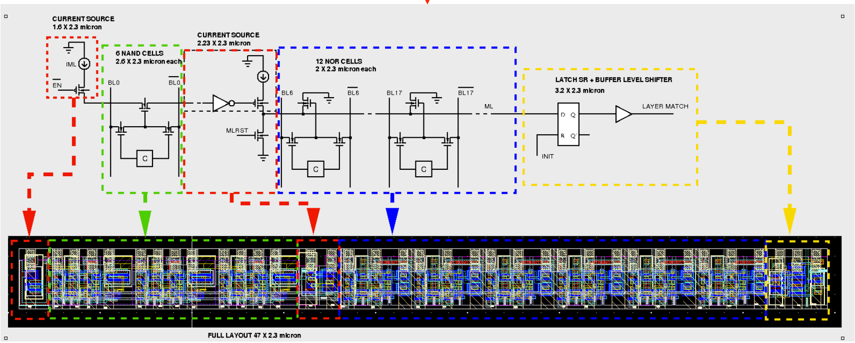
It is a tree of small logic elements that creates a flexible priority encoder.



“AMchip” is an ASIC:

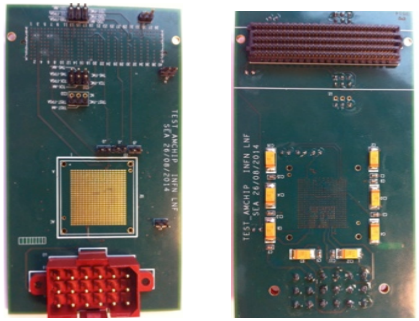
Input: predefined pattern and unknown pattern
output = match/mismatch





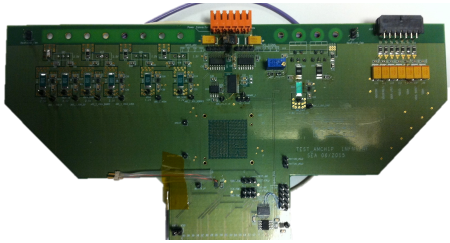
Note: The circuit is not the latest version

Different evaluation of AMchip test bench



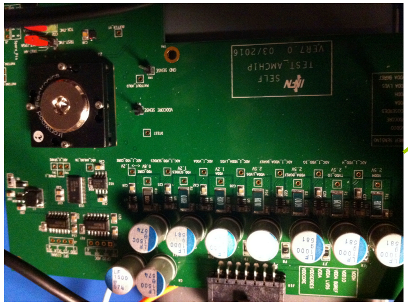
1-The first generation of test-bench

- Have GTX instability and error during the test and data communication
- The comments was sent to the board manufacturer for review of the design
- Temperature monitoring was external
- No on-board ADC was used



2-The second and third generation of test-bench

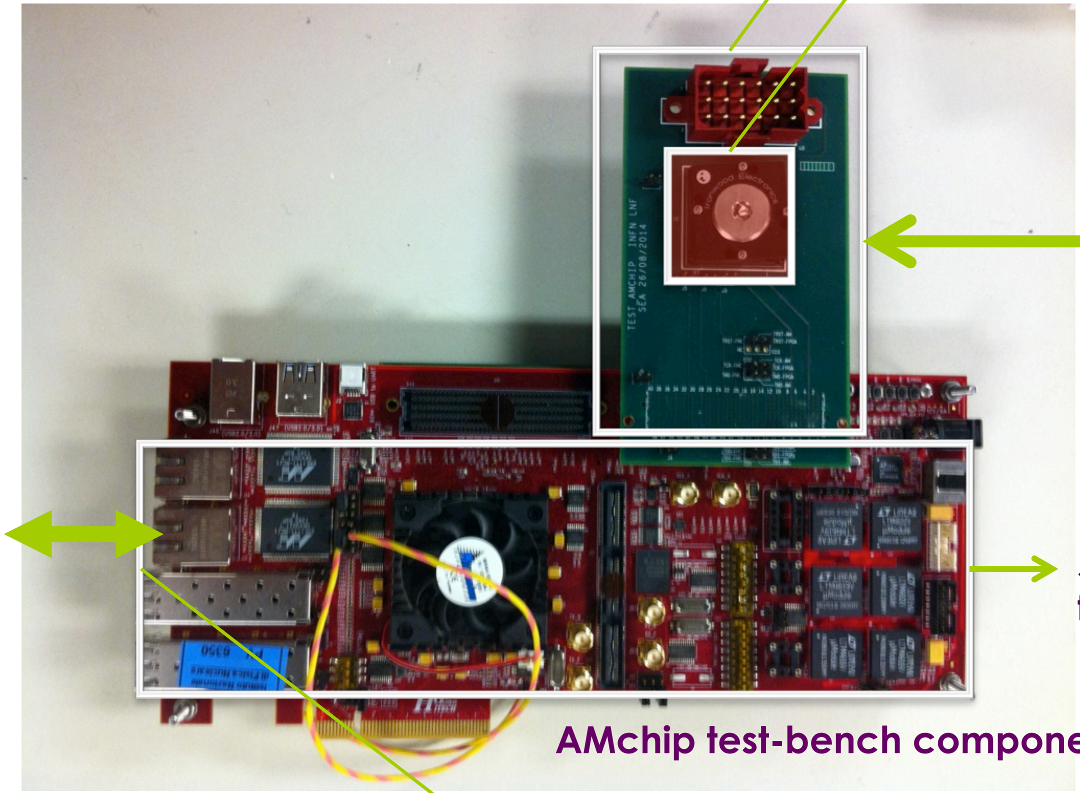
- Power supply problem and some other DC/DC convertors mal-functioning
- Better signal communication and more stable GTX bus
- On-board ADC was included to facilitate real time AMchip monitoring

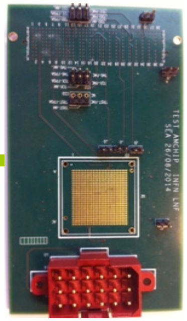


3-The forth generation of test-bench

- Less power supply problem
- Better signal communication and more stable GTX bus
- Some on-board facilities to do AMchip characterization test

Old-test bench based on Virtex-6 Xilinx FPGA





AMchip test board

AMchip

AMchip Pattern in/out data& command (Ethernet)

JTAG interface to download firmware into FPGA

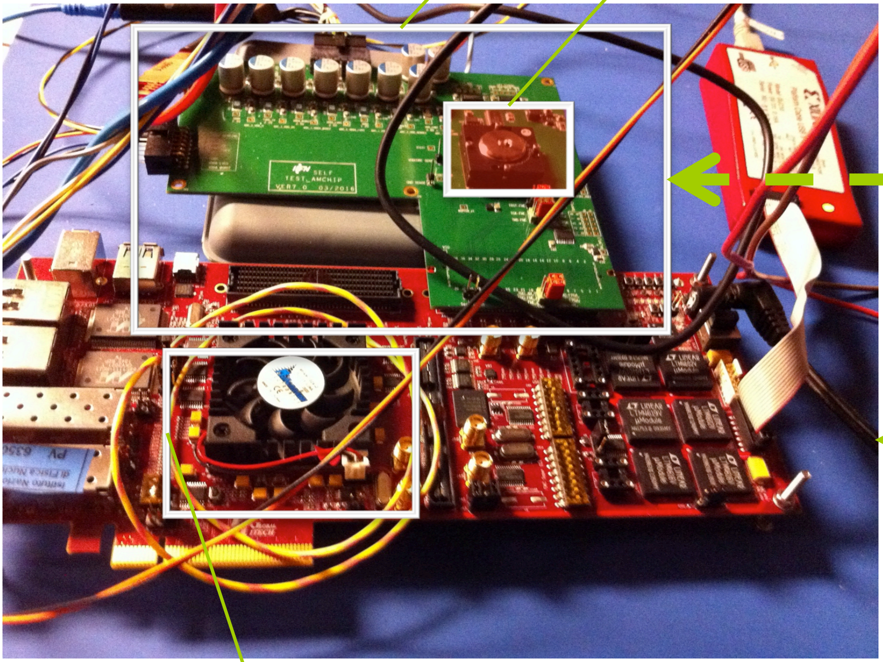
AMchip test-bench components

Xilinx Virtex-6 FPGA

Old-test bench based on Virtex-6 Xilinx FPGA

AMchip test board

AMchip



AMchip Pattern in/out data& command (Ethernet)

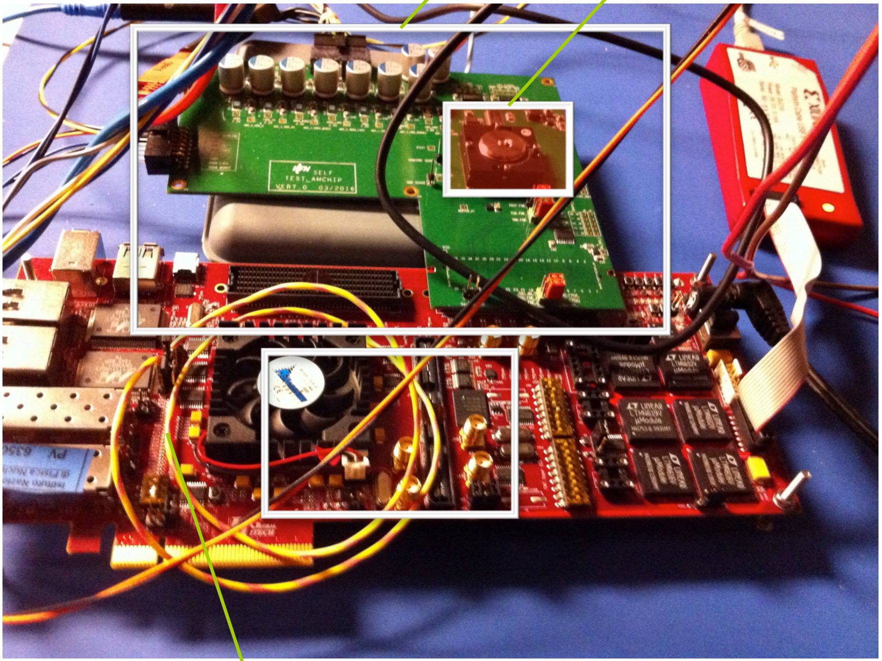
JTAG interface for downloading firmware into FPGA

AMchip test-bench components

Xilinx Virtex-6 FPGA

Old-test bench based on Virtex-6 Xilinx FPGA

AMchip Pattern in/out data& command (Ethernet)

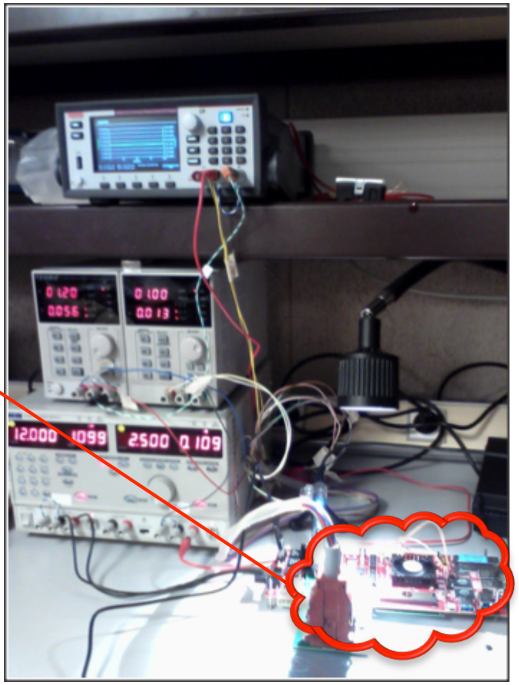


AMchip test board

AMchip

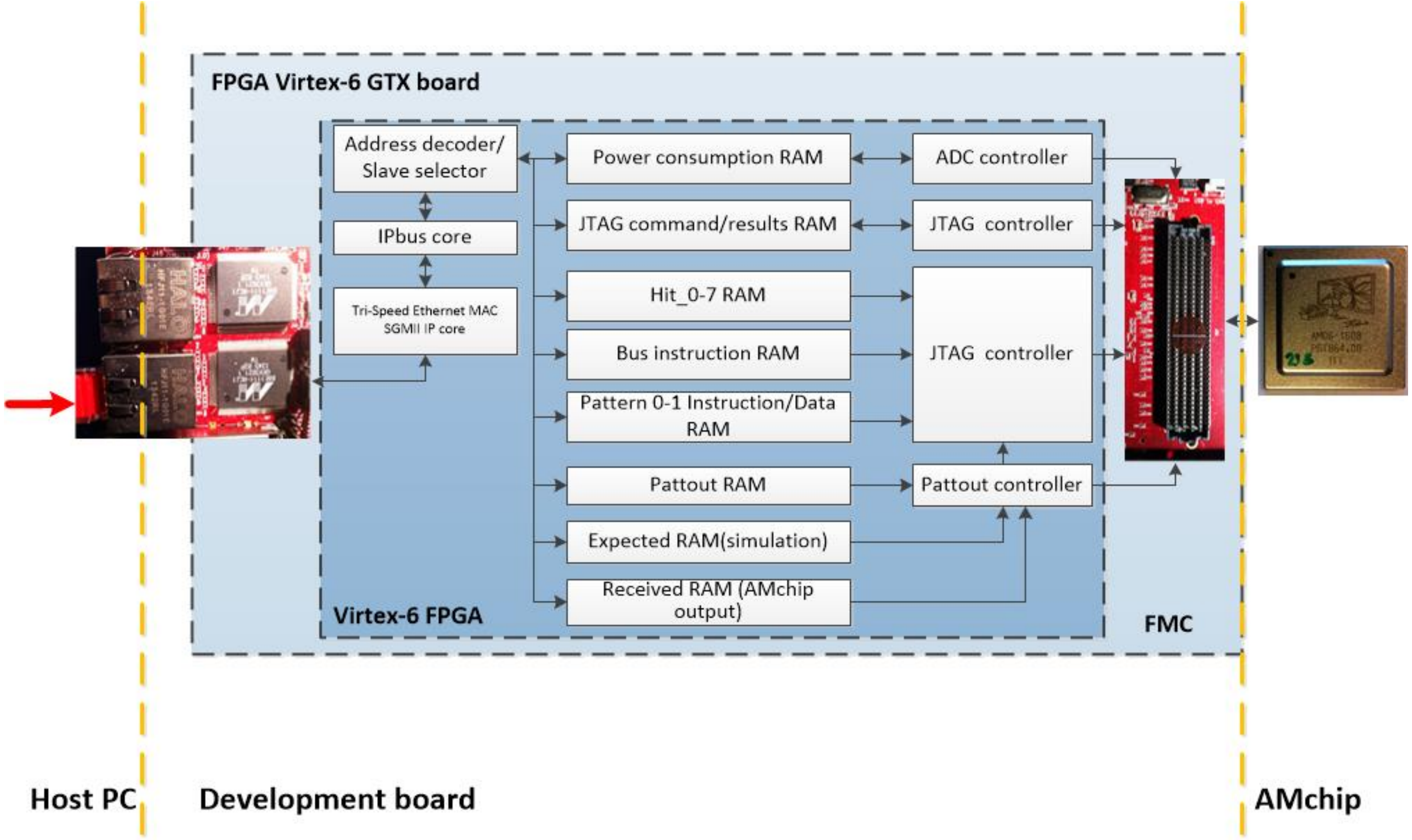
AMchip test-bench components

Xilinx Virtex-6 FPGA

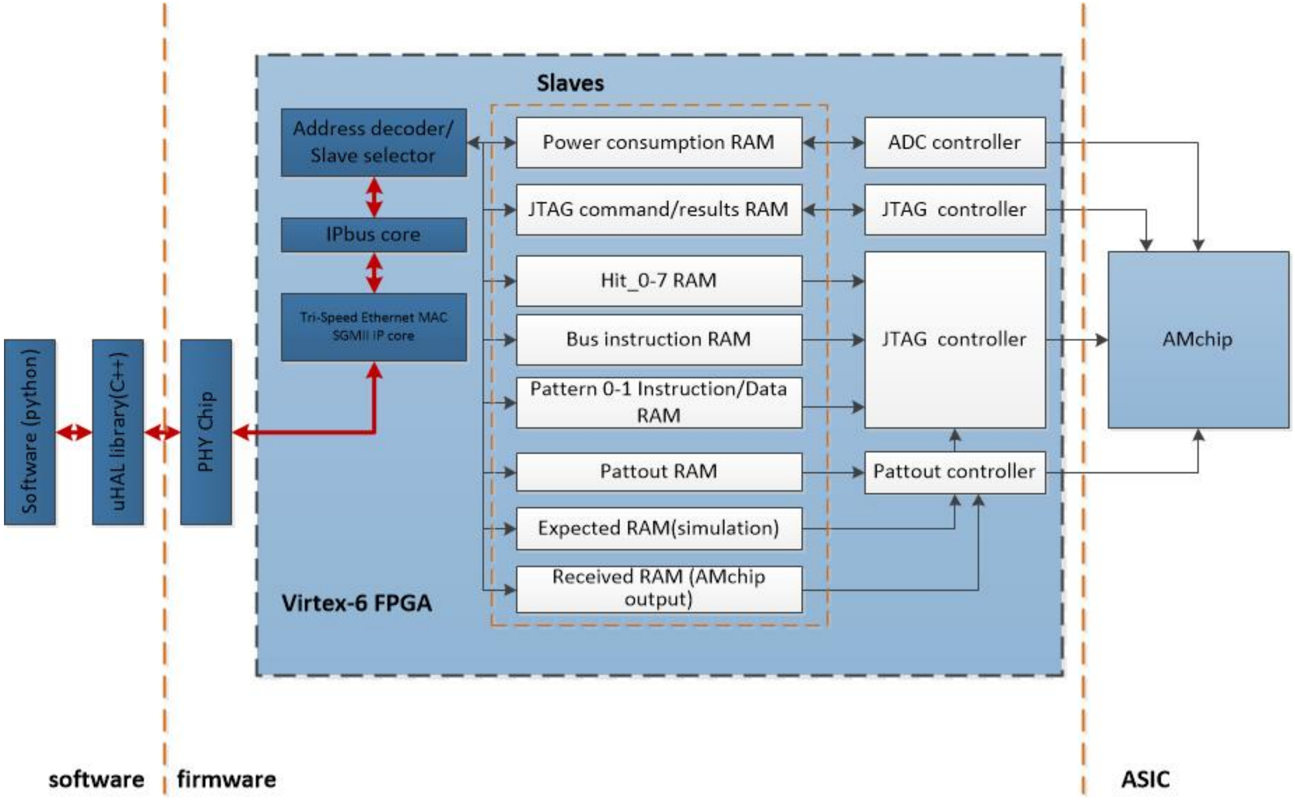


Complete set of test-bench

Firmware and FPGA architecture



FPGA and its interfacing with PC



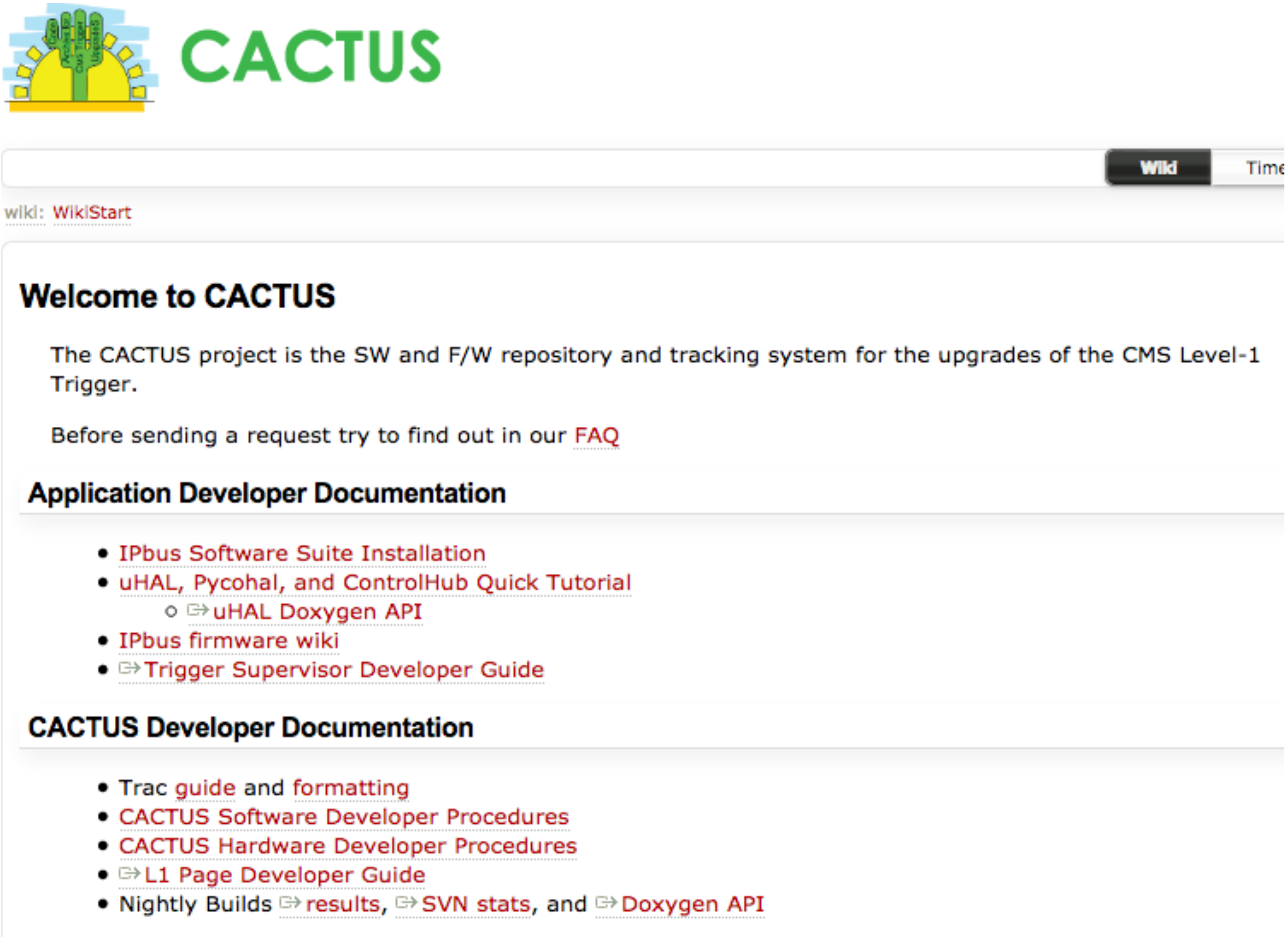
More information about uHAL library can be found here:

https://svnweb.cern.ch/trac/cactus

Further information about IPbus:

https://svnweb.cern.ch/trac/cactus/wiki/uhalQuickTutorial#HowtoInstalltheIPbusSuite

Interfacing with PC through IPbus



wiki: WikiStart

Welcome to CACTUS

The CACTUS project is the SW and F/W repository and tracking system for the upgrades of the CMS Level-1 Trigger.

Before sending a request try to find out in our FAQ

Application Developer Documentation

- IPbus Software Suite Installation
- uHAL, Pycohal, and ControlHub Quick Tutorial
  - ⇨uHAL Doxygen API
- IPbus firmware wiki
- ⇨Trigger Supervisor Developer Guide

CACTUS Developer Documentation

- Trac guide and formatting
- CACTUS Software Developer Procedures
- CACTUS Hardware Developer Procedures
- ⇨L1 Page Developer Guide
- Nightly Builds ⇨results, ⇨SVN stats, and ⇨Doxygen API

Sequence alignment using AMchip

Activities	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16	Mar-16	Apr-16	May-16	Jun-16	Jul-16	Aug-16	Sep-16	Oct-16	Nov-16	Dec-16	Jan-17	Feb-17	Mar-17
AMchip Test-bench I (firmware and software)	🟩	🟩	🟩	🟩	🟩	🟩	🟩																	
Sequence alignment based on AMchip					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩									
IPMC board																								
AMchip Test-bench 2 (firmware and software)																								
presentation and documentstation																								

Sequence matching problem

Research question/answer:

- **How we can align two sequences versus each other?**

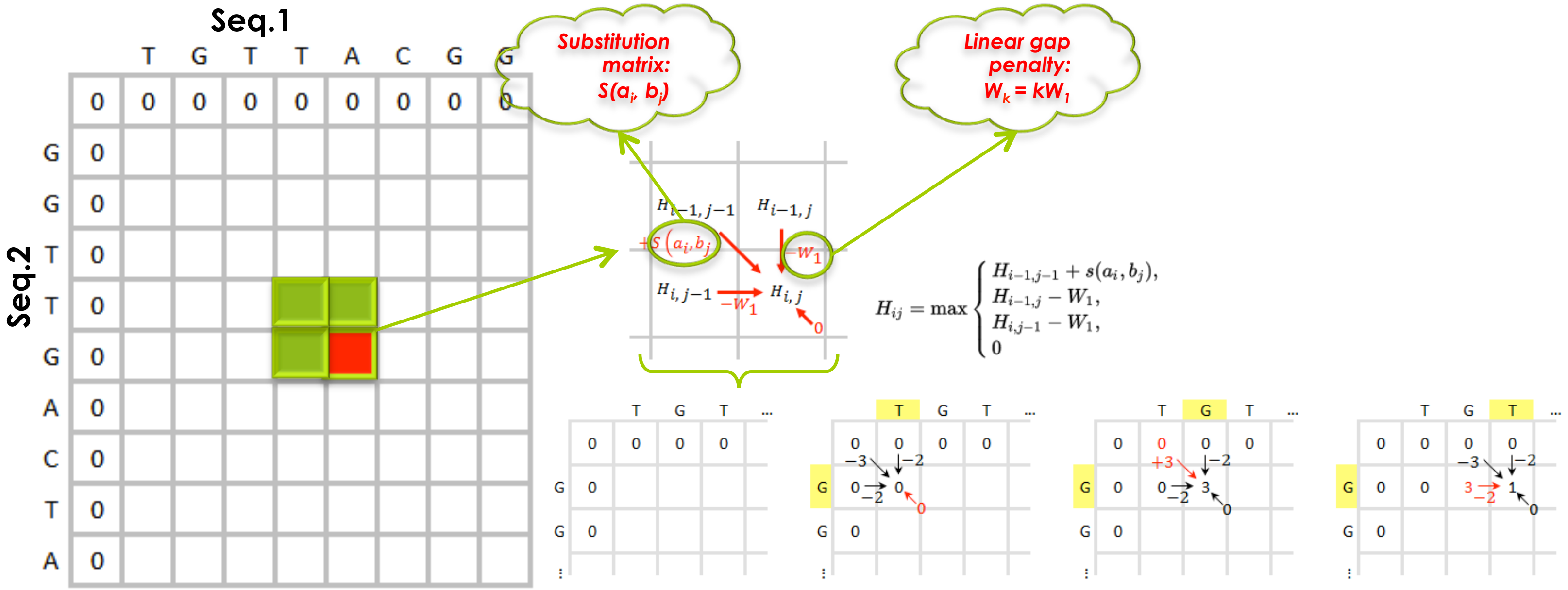
Seq. 1: TGTTACGG
Seq. 2: GGTTGACTA

- **answer: smith-Waterman (SW) algorithm**

Alignment of above two sequences:
-GTT- AC--
-GTTGAC--

- **Steps of optimal SW algorithm:**

1. **Establish a set of rules (penalty weight and substitution matrix)**
2. **Calculate scoring matrix**
3. **Find Absolut maximum**
4. **Trace back from maximum to zero**
5. **Applying alignment rule (insertion, deletion, aligned): start from maximum → end to zero**



Seq.1

Seq.2

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0								
G	0								
T	0								
T	0								
G	0								
A	0								
C	0								
T	0								
A	0								

Substitution matrix: S(a_i, b_j)

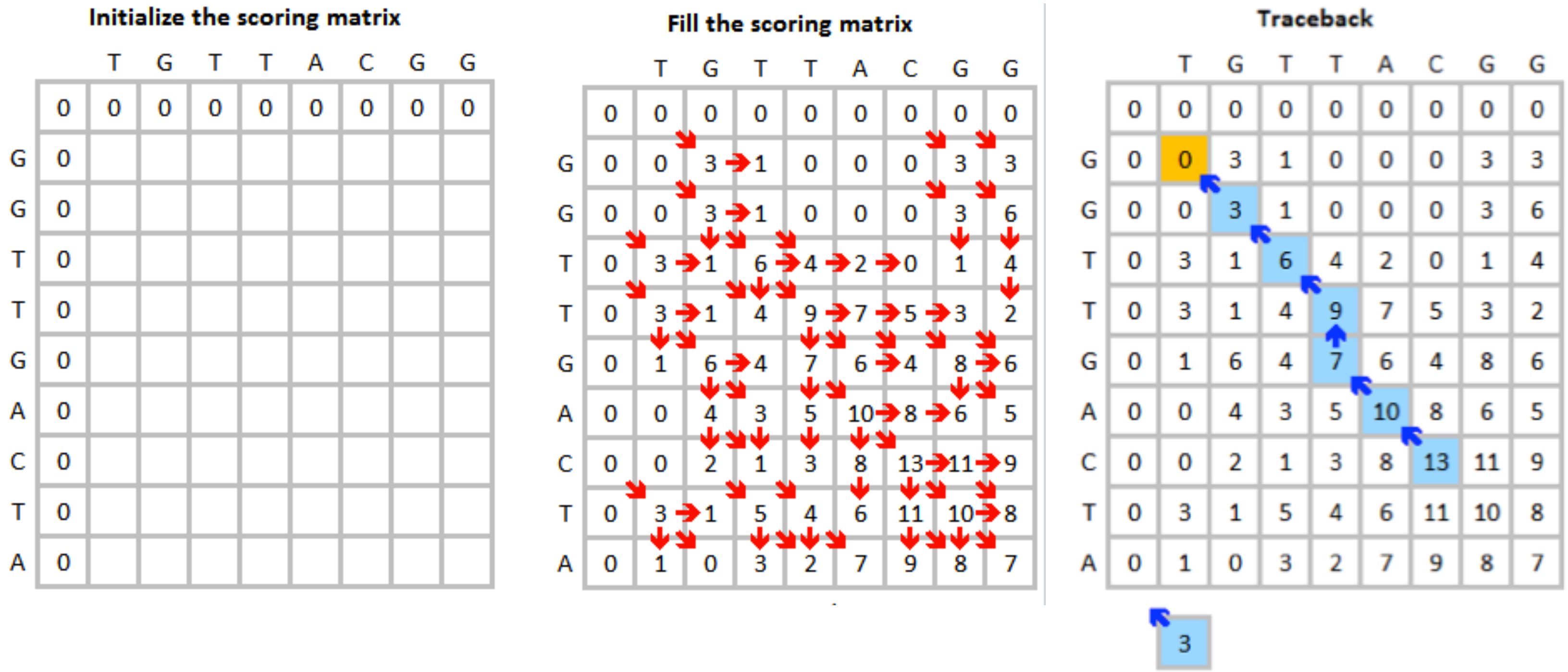
Linear gap penalty: $W_k = kW_1$

$$H_{ij} = \max \begin{cases} H_{i-1,j-1} + s(a_i, b_j), \\ H_{i-1,j} - W_1, \\ H_{i,j-1} - W_1, \\ 0 \end{cases}$$

Sequence matching problem

- **Practice solution based on SW:**

1. Establish a set of rules (penalty weight and substitution matrix)
2. Calculate scoring matrix
3. Finding maximum value in the matrix
4. Tracking back from maximum to zero
5. Applying alignment rule (insertion, deletion, aligned): start from maximum → end to zero



Initialize the scoring matrix

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0								
G	0								
T	0								
T	0								
G	0								
A	0								
C	0								
T	0								
A	0								

Fill the scoring matrix

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	1	0	0	0	3	3
G	0	0	3	1	0	0	0	3	6
T	0	3	1	6	4	2	0	1	4
T	0	3	1	4	9	7	5	3	2
G	0	1	6	4	7	6	4	8	6
A	0	0	4	3	5	10	8	6	5
C	0	0	2	1	3	8	13	11	9
T	0	3	1	5	4	6	11	10	8
A	0	1	0	3	2	7	9	8	7

Traceback

		T	G	T	T	A	C	G	G
	0	0	0	0	0	0	0	0	0
G	0	0	3	1	0	0	0	3	3
G	0	0	3	1	0	0	0	3	6
T	0	3	1	6	4	2	0	1	4
T	0	3	1	4	9	7	5	3	2
G	0	1	6	4	7	6	4	8	6
A	0	0	4	3	5	10	8	6	5
C	0	0	2	1	3	8	13	11	9
T	0	3	1	5	4	6	11	10	8
A	0	1	0	3	2	7	9	8	7

3

```
G T T - A C
| | |   | |
G T T G A C
```

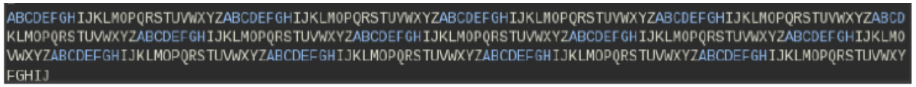
This is called local alignment algorithm.

Sequence matching algorithm

Global and local alignment

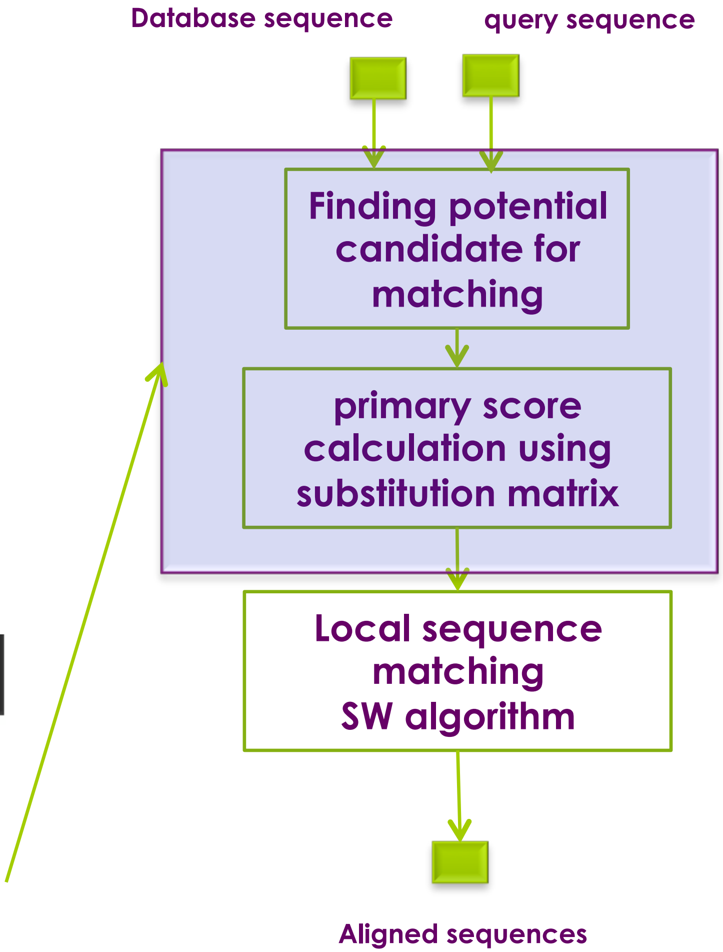
- **Problems:**
  - The database sequences are really long (in some cases might be **1 million characters** long)
  - The **dimension of the matrix** grows very rapidly and the complexity of the calculation increases exponentially
  - The **calculation time** increases (might takes a week)

- **solution:**
  - Global alignment proposed with following principles:
    1. Seeds will be detected along data base sequences
    2. Local sequence matching will be applied only to the selected seeds



```
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
FGHIJ
```

- **Where AMchip can be used?**
  - AMchip can help to find the seeds.
- Selected seeds will be pushed to FPGA for local sequence matching

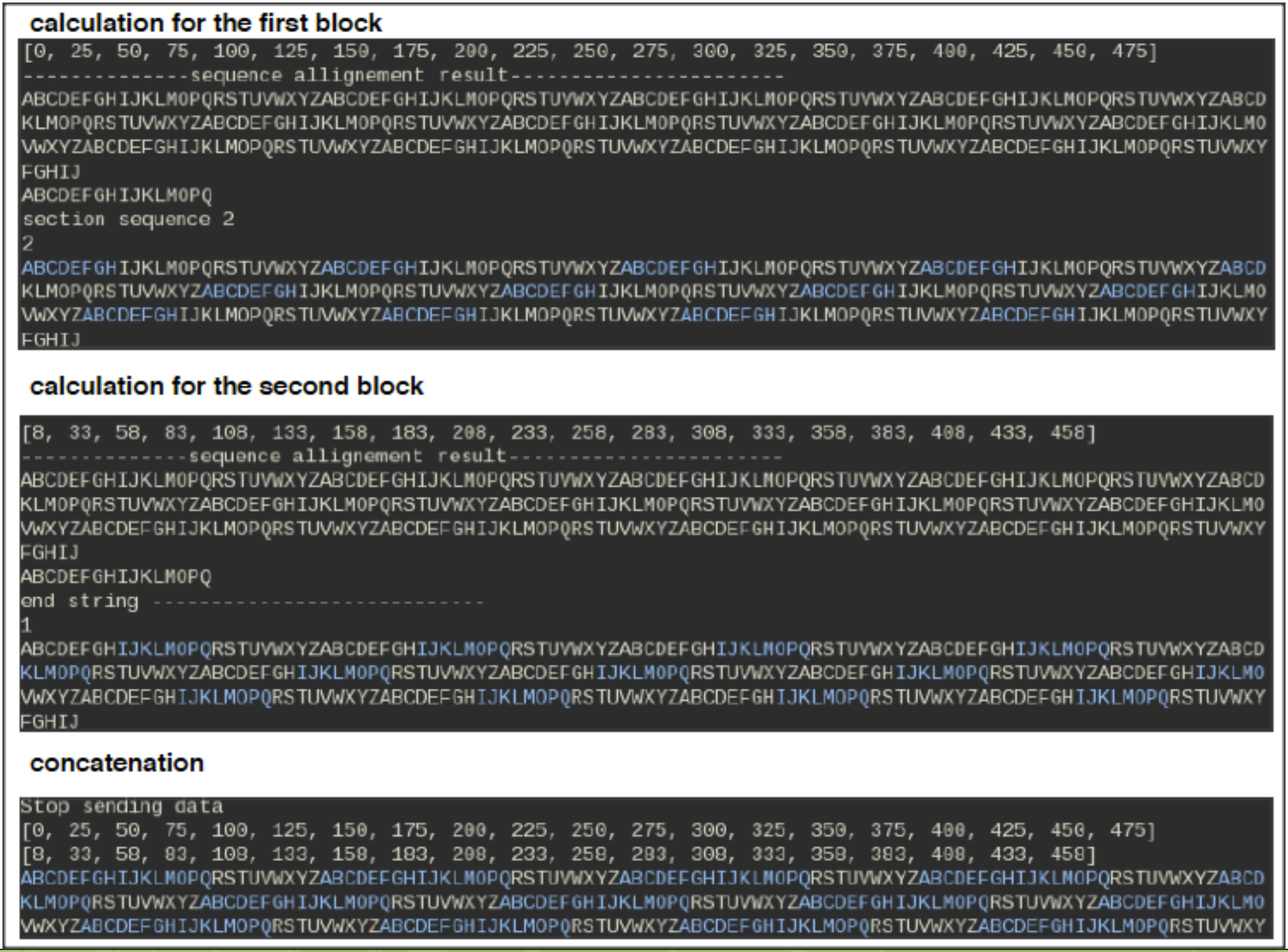

Database sequence
query sequence
Finding potential candidate for matching
primary score calculation using substitution matrix
Local sequence matching SW algorithm
Aligned sequences

- **How AMchip perform seed selection?**

Answer: Pattern matching using convolution operator

AMchip-Sequence alignment

results



calculation for the first block

```
[0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350, 375, 400, 425, 450, 475]
--------------sequence allignement result-----------------------
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
FGHIJ
ABCDEFGHIJKLMOPQ
section sequence 2
2
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
FGHIJ
```

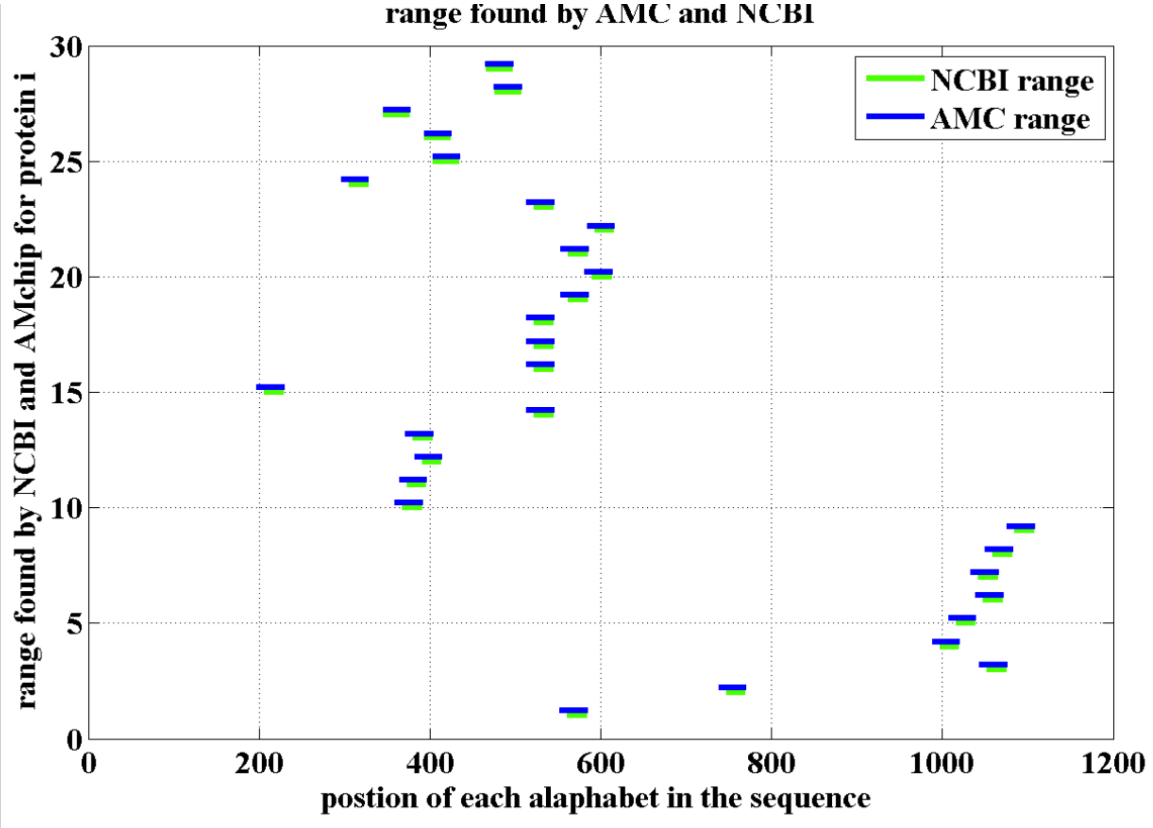
calculation for the second block

```
[8, 33, 58, 83, 108, 133, 158, 183, 208, 233, 258, 283, 308, 333, 358, 383, 408, 433, 458]
--------------sequence allignement result-----------------------
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
FGHIJ
ABCDEFGHIJKLMOPQ
end string ---------------------------
1
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
FGHIJ
```

concatenation

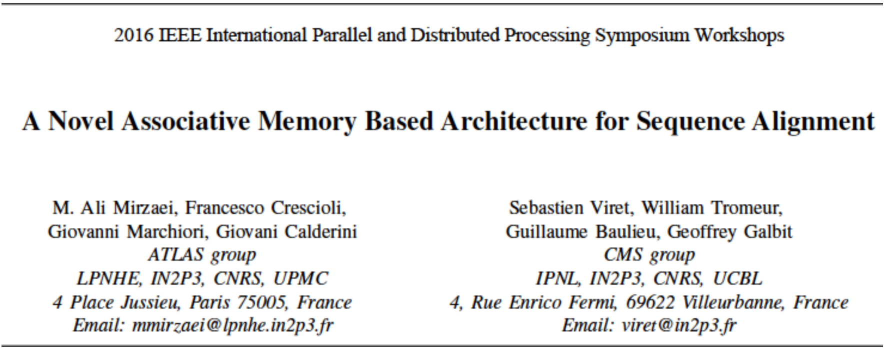
```
Stop sending data
[0, 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 325, 350, 375, 400, 425, 450, 475]
[8, 33, 58, 83, 108, 133, 158, 183, 208, 233, 258, 283, 308, 333, 358, 383, 408, 433, 458]
ABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCD
KLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMO
VWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXYZABCDEFGHIJKLMOPQRSTUVWXY
```

Result and publication



HiCOMB 2016

15th IEEE International Workshop on High Performance Computational Biology
May 22-23, 2016
Chicago, Illinois, USA



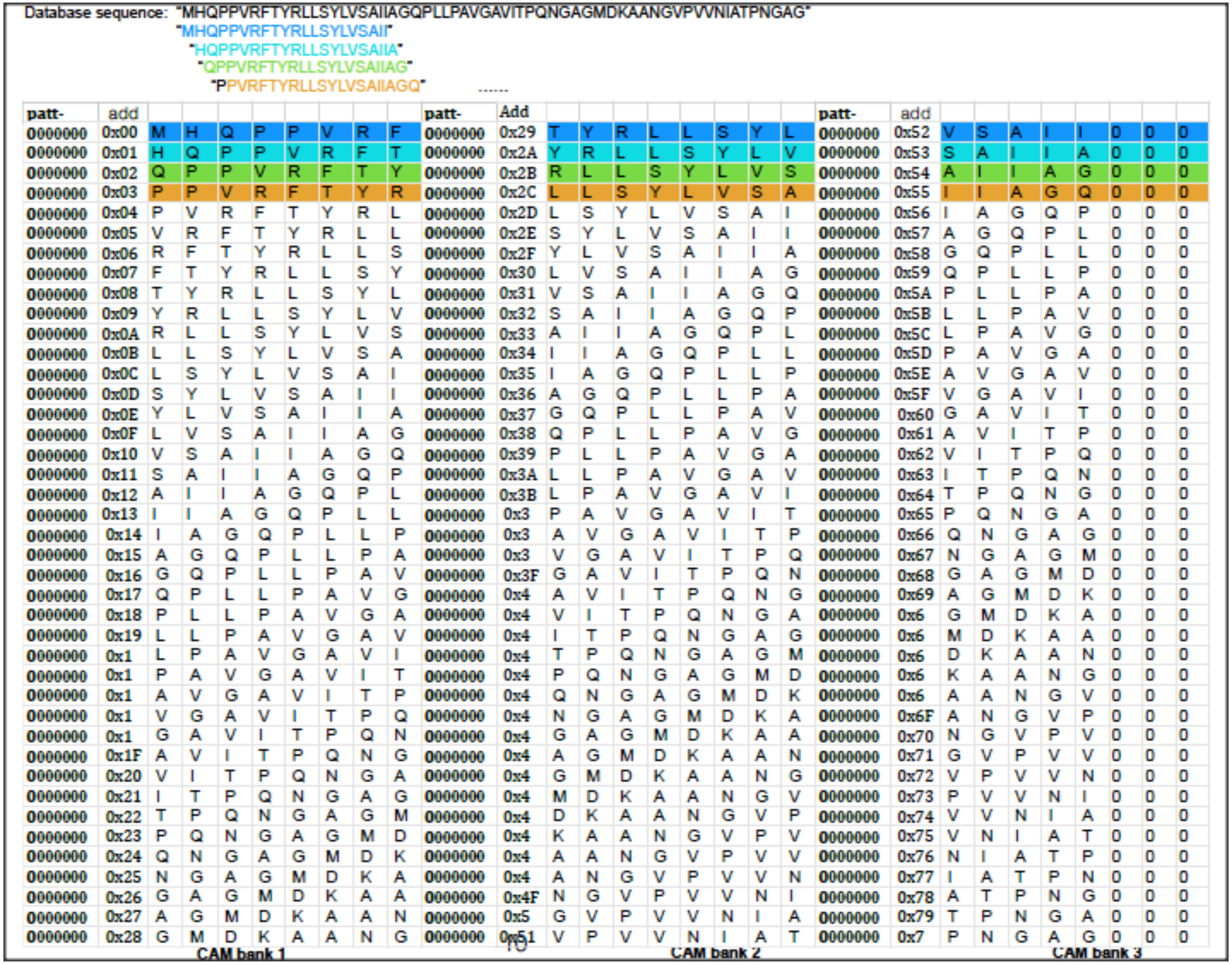
2016 IEEE International Parallel and Distributed Processing Symposium Workshops

A Novel Associative Memory Based Architecture for Sequence Alignment

M. Ali Mirzaei, Francesco Crescioli,
Giovanni Marchiori, Giovani Calderini
ATLAS group
LPNHE, IN2P3, CNRS, UPMC
4 Place Jussieu, Paris 75005, France
Email: mmirzaei@lpnhe.in2p3.fr

Sebastien Viret, William Tromeur,
Guillaume Baulieu, Geoffrey Galbit
CMS group
IPNL, IN2P3, CNRS, UCBL
4, Rue Enrico Fermi, 69622 Villeurbanne, France
Email: viret@in2p3.fr

Reformulation of the problem for AMchip



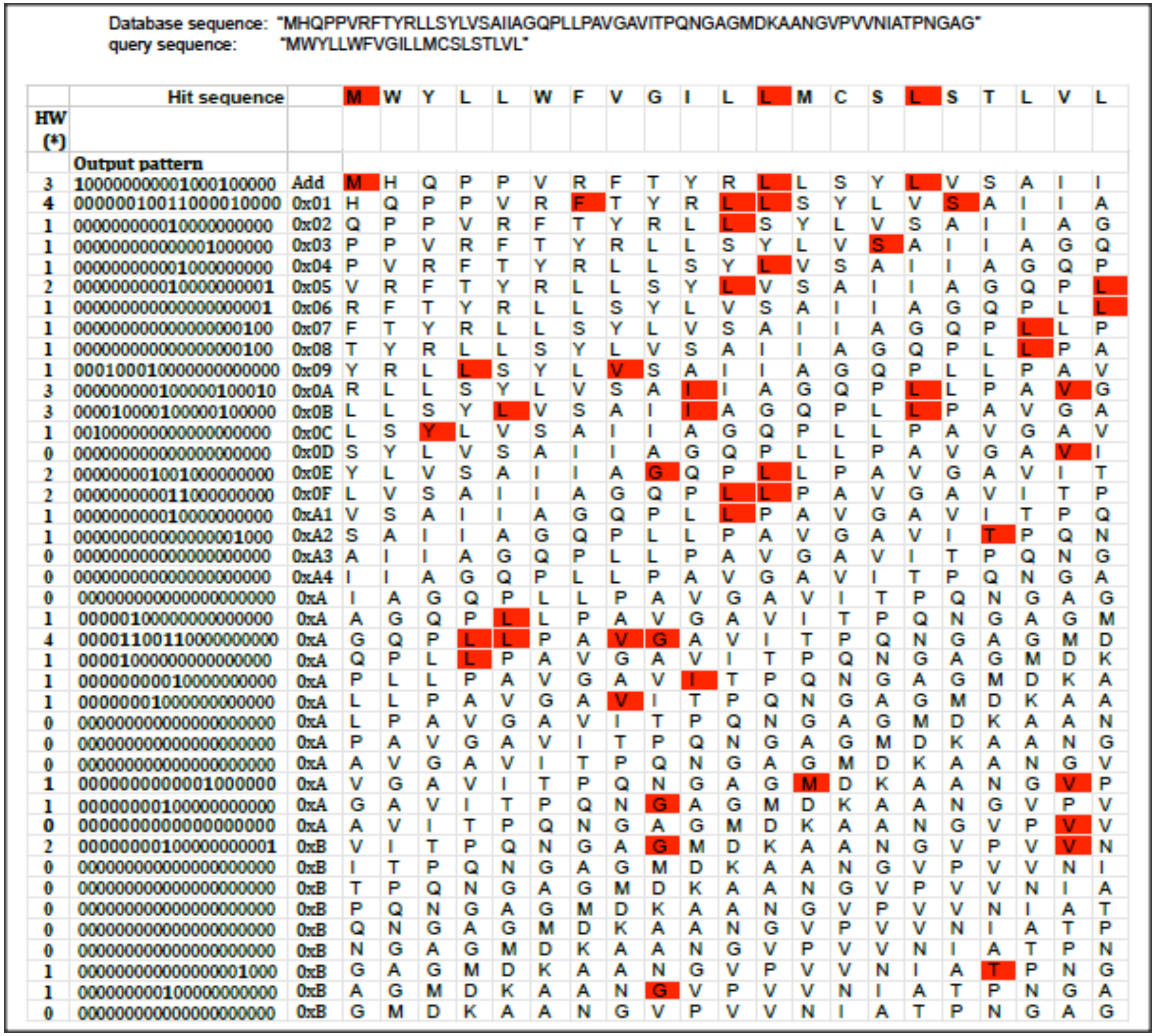
Database sequence: "MHQPPVRFTYRLLSYLVSAIIAGQPLLPAVGAVITPQNGAGMDKAANGVPVVNIATPNGAG"

"MHQPPVRFTYRLLSYLVSAII"
"HQPPVRFTYRLLSYLVSAIIA"
"QPPVRFTYRLLSYLVSAIIAG"
"PPVRFTYRLLSYLVSAIIAGQ" ......

patt-	add									patt-	Add									patt-	add								
0000000	0x00	M	H	Q	P	P	V	R	F	0000000	0x29	T	Y	R	L	L	S	Y	L	0000000	0x52	V	S	A	I	I	0	0	0
0000000	0x01	H	Q	P	P	V	R	F	T	0000000	0x2A	Y	R	L	L	S	Y	L	V	0000000	0x53	S	A	I	I	A	0	0	0
0000000	0x02	Q	P	P	V	R	F	T	Y	0000000	0x2B	R	L	L	S	Y	L	V	S	0000000	0x54	A	I	I	A	G	0	0	0
0000000	0x03	P	P	V	R	F	T	Y	R	0000000	0x2C	L	L	S	Y	L	V	S	A	0000000	0x55	I	I	A	G	Q	0	0	0
0000000	0x04	P	V	R	F	T	Y	R	L	0000000	0x2D	L	S	Y	L	V	S	A	I	0000000	0x56	I	A	G	Q	P	0	0	0
0000000	0x05	V	R	F	T	Y	R	L	L	0000000	0x2E	S	Y	L	V	S	A	I	I	0000000	0x57	A	G	Q	P	L	0	0	0
0000000	0x06	R	F	T	Y	R	L	L	S	0000000	0x2F	Y	L	V	S	A	I	I	A	0000000	0x58	G	Q	P	L	L	0	0	0
0000000	0x07	F	T	Y	R	L	L	S	Y	0000000	0x30	L	V	S	A	I	I	A	G	0000000	0x59	Q	P	L	L	P	0	0	0
0000000	0x08	T	Y	R	L	L	S	Y	L	0000000	0x31	V	S	A	I	I	A	G	Q	0000000	0x5A	P	L	L	P	A	0	0	0
0000000	0x09	Y	R	L	L	S	Y	L	V	0000000	0x32	S	A	I	I	A	G	Q	P	0000000	0x5B	L	L	P	A	V	0	0	0
0000000	0x0A	R	L	L	S	Y	L	V	S	0000000	0x33	A	I	I	A	G	Q	P	L	0000000	0x5C	L	P	A	V	G	0	0	0
0000000	0x0B	L	L	S	Y	L	V	S	A	0000000	0x34	I	I	A	G	Q	P	L	L	0000000	0x5D	P	A	V	G	A	0	0	0
0000000	0x0C	L	S	Y	L	V	S	A	I	0000000	0x35	I	A	G	Q	P	L	L	P	0000000	0x5E	A	V	G	A	V	0	0	0
0000000	0x0D	S	Y	L	V	S	A	I	I	0000000	0x36	A	G	Q	P	L	L	P	A	0000000	0x5F	V	G	A	V	I	0	0	0
0000000	0x0E	Y	L	V	S	A	I	I	A	0000000	0x37	G	Q	P	L	L	P	A	V	0000000	0x60	G	A	V	I	T	0	0	0
0000000	0x0F	L	V	S	A	I	I	A	G	0000000	0x38	Q	P	L	L	P	A	V	G	0000000	0x61	A	V	I	T	P	0	0	0
0000000	0x10	V	S	A	I	I	A	G	Q	0000000	0x39	P	L	L	P	A	V	G	A	0000000	0x62	V	I	T	P	Q	0	0	0
0000000	0x11	S	A	I	I	A	G	Q	P	0000000	0x3A	L	L	P	A	V	G	A	V	0000000	0x63	I	T	P	Q	N	0	0	0
0000000	0x12	A	I	I	A	G	Q	P	L	0000000	0x3B	L	P	A	V	G	A	V	I	0000000	0x64	T	P	Q	N	G	0	0	0
0000000	0x13	I	I	A	G	Q	P	L	L	0000000	0x3	P	A	V	G	A	V	I	T	0000000	0x65	P	Q	N	G	A	0	0	0
0000000	0x14	I	A	G	Q	P	L	L	P	0000000	0x3	A	V	G	A	V	I	T	P	0000000	0x66	Q	N	G	A	G	0	0	0
0000000	0x15	A	G	Q	P	L	L	P	A	0000000	0x3	V	G	A	V	I	T	P	Q	0000000	0x67	N	G	A	G	M	0	0	0
0000000	0x16	G	Q	P	L	L	P	A	V	0000000	0x3F	G	A	V	I	T	P	Q	N	0000000	0x68	G	A	G	M	D	0	0	0
0000000	0x17	Q	P	L	L	P	A	V	G	0000000	0x4	A	V	I	T	P	Q	N	G	0000000	0x69	A	G	M	D	K	0	0	0
0000000	0x18	P	L	L	P	A	V	G	A	0000000	0x4	V	I	T	P	Q	N	G	A	0000000	0x6	G	M	D	K	A	0	0	0
0000000	0x19	L	L	P	A	V	G	A	V	0000000	0x4	I	T	P	Q	N	G	A	G	0000000	0x6	M	D	K	A	A	0	0	0
0000000	0x1	L	P	A	V	G	A	V	I	0000000	0x4	T	P	Q	N	G	A	G	M	0000000	0x6	D	K	A	A	N	0	0	0
0000000	0x1	P	A	V	G	A	V	I	T	0000000	0x4	P	Q	N	G	A	G	M	D	0000000	0x6	K	A	A	N	G	0	0	0
0000000	0x1	A	V	G	A	V	I	T	P	0000000	0x4	Q	N	G	A	G	M	D	K	0000000	0x6	A	A	N	G	V	0	0	0
0000000	0x1	V	G	A	V	I	T	P	Q	0000000	0x4	N	G	A	G	M	D	K	A	0000000	0x6F	A	N	G	V	P	0	0	0
0000000	0x1	G	A	V	I	T	P	Q	N	0000000	0x4	G	A	G	M	D	K	A	A	0000000	0x70	N	G	V	P	V	0	0	0
0000000	0x1F	A	V	I	T	P	Q	N	G	0000000	0x4	A	G	M	D	K	A	A	N	0000000	0x71	G	V	P	V	V	0	0	0
0000000	0x20	V	I	T	P	Q	N	G	A	0000000	0x4	G	M	D	K	A	A	N	G	0000000	0x72	V	P	V	V	N	0	0	0
0000000	0x21	I	T	P	Q	N	G	A	G	0000000	0x4	M	D	K	A	A	N	G	V	0000000	0x73	P	V	V	N	I	0	0	0
0000000	0x22	T	P	Q	N	G	A	G	M	0000000	0x4	D	K	A	A	N	G	V	P	0000000	0x74	V	V	N	I	A	0	0	0
0000000	0x23	P	Q	N	G	A	G	M	D	0000000	0x4	K	A	A	N	G	V	P	V	0000000	0x75	V	N	I	A	T	0	0	0
0000000	0x24	Q	N	G	A	G	M	D	K	0000000	0x4	A	A	N	G	V	P	V	V	0000000	0x76	N	I	A	T	P	0	0	0
0000000	0x25	N	G	A	G	M	D	K	A	0000000	0x4	A	N	G	V	P	V	V	N	0000000	0x77	I	A	T	P	N	0	0	0
0000000	0x26	G	A	G	M	D	K	A	A	0000000	0x4F	N	G	V	P	V	V	N	I	0000000	0x78	A	T	P	N	G	0	0	0
0000000	0x27	A	G	M	D	K	A	A	N	0000000	0x5	G	V	P	V	V	N	I	A	0000000	0x79	T	P	N	G	A	0	0	0
0000000	0x28	G	M	D	K	A	A	N	G	0000000	0x51	V	P	V	V	N	I	A	T	0000000	0x7	P	N	G	A	G	0	0	0

CAM bank 1 CAM bank 2 CAM bank 3

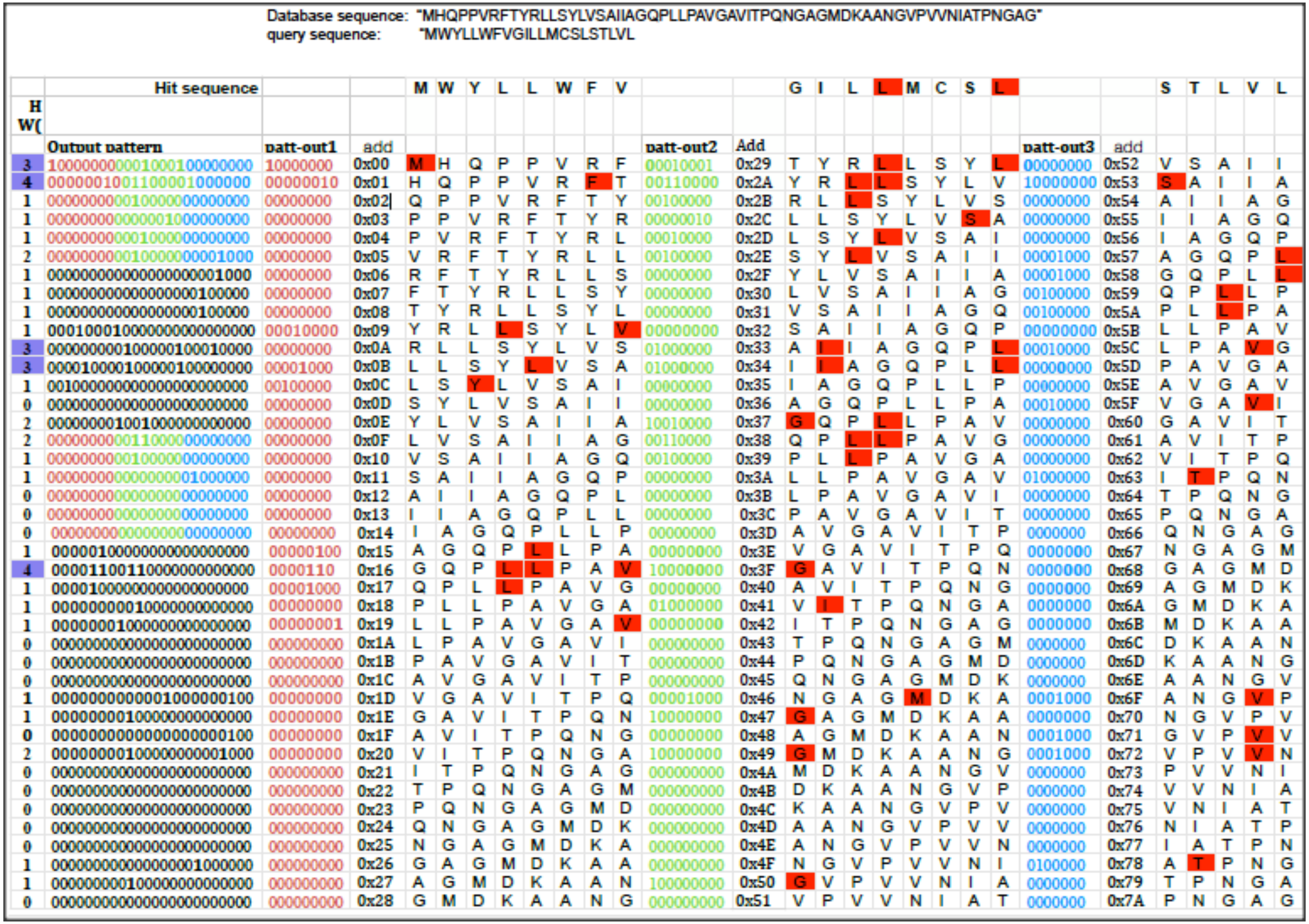
Reformulation of the problem for AMchip



Database sequence: "MHQPPVRFTYRLLSYLVSAIIAGQPLLPAVGAVITPQNGAGMDKAANGVPVVNIATPNGAG"
query sequence: "MWYLLWFVGILLMCSLSTLVL"

	Hit sequence		M	W	Y	L	L	W	F	V	G	I	L	L	M	C	S	L	S	T	L	V	L
HW (*)																							
	Output pattern																						
3	1000000000001000100000	Add	M	H	Q	P	P	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I
4	000000100110000010000	0x01	H	Q	P	P	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A
1	000000000010000000000	0x02	Q	P	P	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G
1	000000000000001000000	0x03	P	P	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q
1	000000000001000000000	0x04	P	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P
2	000000000010000000001	0x05	V	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L
1	000000000000000000001	0x06	R	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L
1	000000000000000000100	0x07	F	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P
1	000000000000000000100	0x08	T	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A
1	000100010000000000000	0x09	Y	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V
3	000000001000001000010	0x0A	R	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G
3	000010000100000100000	0x0B	L	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A
1	001000000000000000000	0x0C	L	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V
0	000000000000000000000	0x0D	S	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I
2	000000001001000000000	0x0E	Y	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T
2	000000000011000000000	0x0F	L	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P
1	000000000010000000000	0xA1	V	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q
1	000000000000000001000	0xA2	S	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N
0	000000000000000000000	0xA3	A	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G
0	000000000000000000000	0xA4	I	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A
0	000000000000000000000	0xA	I	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G
1	000001000000000000000	0xA	A	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M
4	000011001100000000000	0xA	G	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D
1	000010000000000000000	0xA	Q	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K
1	000000000100000000000	0xA	P	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A
1	000000010000000000000	0xA	L	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A
0	000000000000000000000	0xA	L	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N
0	000000000000000000000	0xA	P	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G
0	000000000000000000000	0xA	A	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V
1	000000000000001000000	0xA	V	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P
1	000000001000000000000	0xA	G	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V
0	000000000000000000000	0xA	A	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V
2	000000001000000000001	0xB	V	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N
0	000000000000000000000	0xB	I	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I
0	000000000000000000000	0xB	T	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A
0	000000000000000000000	0xB	P	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T
0	000000000000000000000	0xB	Q	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T	P
0	000000000000000000000	0xB	N	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T	P	N
1	000000000000000001000	0xB	G	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T	P	N	G
1	000000000100000000000	0xB	A	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T	P	N	G	A
0	000000000000000000000	0xB	G	M	D	K	A	A	N	G	V	P	V	V	N	I	A	T	P	N	G	A	G

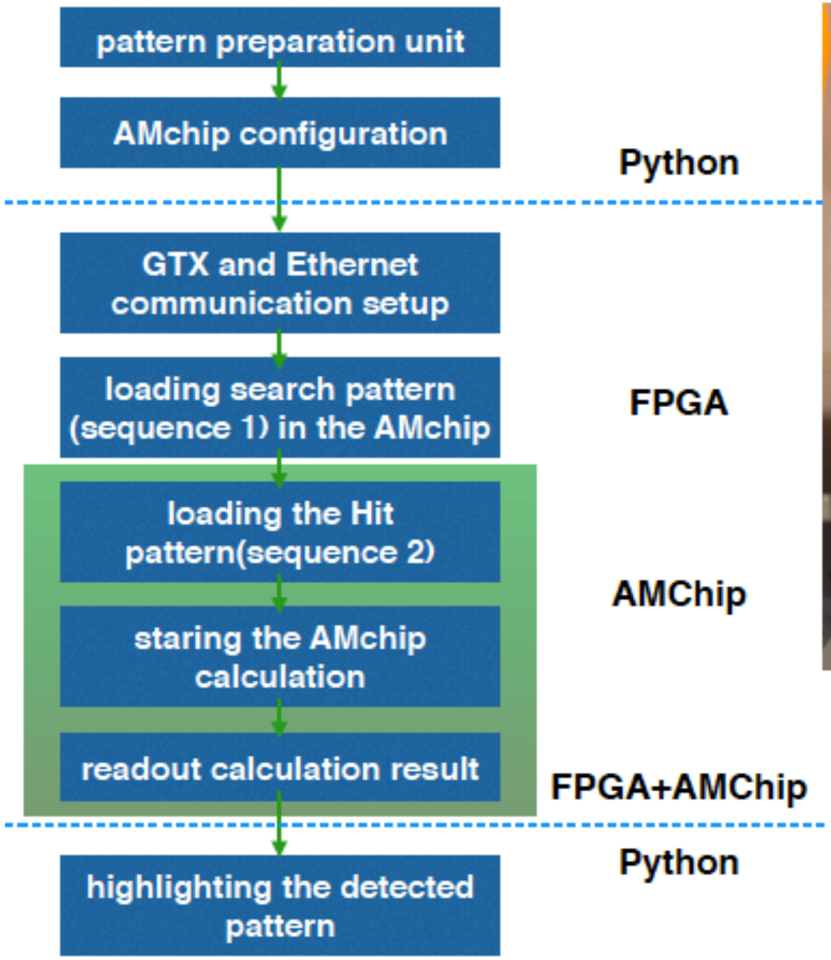
Reformulation of the problem for AMchip



Database sequence: "MHQPPVRFTYRLLSYLVSAIIAGQPLLPAVGAVITPQNGAGMDKAANGVPVVNIATPNGAG"
query sequence: "MWYLLWFVGILLMCSLSTLVL

	Hit sequence			M	W	Y	L	L	W	F	V			G	I	L	L	M	C	S	L			S	T	L	V	L
H W(																												
	Output pattern	patt-out1	add									patt-out2	Add									patt-out3	add					
3	100000000001000100000000	10000000	0x00	M	H	Q	P	P	V	R	F	00010001	0x29	T	Y	R	L	L	S	Y	L	00000000	0x52	V	S	A	I	I
4	0000001001100001000000	00000010	0x01	H	Q	P	P	V	R	F	T	00110000	0x2A	Y	R	L	L	S	Y	L	V	10000000	0x53	S	A	I	I	A
1	000000000010000000000000	00000000	0x02	Q	P	P	V	R	F	T	Y	00100000	0x2B	R	L	L	S	Y	L	V	S	00000000	0x54	A	I	I	A	G
1	000000000000001000000000	00000000	0x03	P	P	V	R	F	T	Y	R	00000010	0x2C	L	L	S	Y	L	V	S	A	00000000	0x55	I	I	A	G	Q
1	000000000001000000000000	00000000	0x04	P	V	R	F	T	Y	R	L	00010000	0x2D	L	S	Y	L	V	S	A	I	00000000	0x56	I	A	G	Q	P
2	000000000010000000001000	00000000	0x05	V	R	F	T	Y	R	L	L	00100000	0x2E	S	Y	L	V	S	A	I	I	00001000	0x57	A	G	Q	P	L
1	000000000000000000001000	00000000	0x06	R	F	T	Y	R	L	L	S	00000000	0x2F	Y	L	V	S	A	I	I	A	00001000	0x58	G	Q	P	L	L
1	000000000000000000100000	00000000	0x07	F	T	Y	R	L	L	S	Y	00000000	0x30	L	V	S	A	I	I	A	G	00100000	0x59	Q	P	L	L	P
1	000000000000000000100000	00000000	0x08	T	Y	R	L	L	S	Y	L	00000000	0x31	V	S	A	I	I	A	G	Q	00100000	0x5A	P	L	L	P	A
1	000100010000000000000000	00010000	0x09	Y	R	L	L	S	Y	L	V	00000000	0x32	S	A	I	I	A	G	Q	P	000000000	0x5B	L	L	P	A	V
3	000000000100000100010000	00000000	0x0A	R	L	L	S	Y	L	V	S	01000000	0x33	A	I	I	A	G	Q	P	L	00010000	0x5C	L	P	A	V	G
3	000010000100000100000000	00001000	0x0B	L	L	S	Y	L	V	S	A	01000000	0x34	I	I	A	G	Q	P	L	L	00000000	0x5D	P	A	V	G	A
1	001000000000000000000000	00100000	0x0C	L	S	Y	L	V	S	A	I	00000000	0x35	I	A	G	Q	P	L	L	P	00000000	0x5E	A	V	G	A	V
0	000000000000000000000000	00000000	0x0D	S	Y	L	V	S	A	I	I	00000000	0x36	A	G	Q	P	L	L	P	A	00010000	0x5F	V	G	A	V	I
2	000000001001000000000000	00000000	0x0E	Y	L	V	S	A	I	I	A	10010000	0x37	G	Q	P	L	L	P	A	V	00000000	0x60	G	A	V	I	T
2	000000000011000000000000	00000000	0x0F	L	V	S	A	I	I	A	G	00110000	0x38	Q	P	L	L	P	A	V	G	00000000	0x61	A	V	I	T	P
1	000000000010000000000000	00000000	0x10	V	S	A	I	I	A	G	Q	00100000	0x39	P	L	L	P	A	V	G	A	00000000	0x62	V	I	T	P	Q
1	000000000000000001000000	00000000	0x11	S	A	I	I	A	G	Q	P	00000000	0x3A	L	L	P	A	V	G	A	V	01000000	0x63	I	T	P	Q	N
0	000000000000000000000000	00000000	0x12	A	I	I	A	G	Q	P	L	00000000	0x3B	L	P	A	V	G	A	V	I	00000000	0x64	T	P	Q	N	G
0	000000000000000000000000	00000000	0x13	I	I	A	G	Q	P	L	L	00000000	0x3C	P	A	V	G	A	V	I	T	00000000	0x65	P	Q	N	G	A
0	000000000000000000000000	00000000	0x14	I	A	G	Q	P	L	L	P	00000000	0x3D	A	V	G	A	V	I	T	P	0000000	0x66	Q	N	G	A	G
1	000001000000000000000000	00000100	0x15	A	G	Q	P	L	L	P	A	00000000	0x3E	V	G	A	V	I	T	P	Q	0000000	0x67	N	G	A	G	M
4	000011001100000000000000	0000110	0x16	G	Q	P	L	L	P	A	V	10000000	0x3F	G	A	V	I	T	P	Q	N	0000000	0x68	G	A	G	M	D
1	000010000000000000000000	00001000	0x17	Q	P	L	L	P	A	V	G	00000000	0x40	A	V	I	T	P	Q	N	G	0000000	0x69	A	G	M	D	K
1	000000000100000000000000	00000000	0x18	P	L	L	P	A	V	G	A	01000000	0x41	V	I	T	P	Q	N	G	A	0000000	0x6A	G	M	D	K	A
1	000000010000000000000000	00000001	0x19	L	L	P	A	V	G	A	V	00000000	0x42	I	T	P	Q	N	G	A	G	0000000	0x6B	M	D	K	A	A
0	000000000000000000000000	000000000	0x1A	L	P	A	V	G	A	V	I	000000000	0x43	T	P	Q	N	G	A	G	M	0000000	0x6C	D	K	A	A	N
0	000000000000000000000000	000000000	0x1B	P	A	V	G	A	V	I	T	000000000	0x44	P	Q	N	G	A	G	M	D	0000000	0x6D	K	A	A	N	G
0	000000000000000000000000	000000000	0x1C	A	V	G	A	V	I	T	P	000000000	0x45	Q	N	G	A	G	M	D	K	0000000	0x6E	A	A	N	G	V
1	000000000000100000000100	00000000	0x1D	V	G	A	V	I	T	P	Q	00001000	0x46	N	G	A	G	M	D	K	A	0001000	0x6F	A	N	G	V	P
1	000000001000000000000000	00000000	0x1E	G	A	V	I	T	P	Q	N	10000000	0x47	G	A	G	M	D	K	A	A	0000000	0x70	N	G	V	P	V
0	000000000000000000000100	00000000	0x1F	A	V	I	T	P	Q	N	G	00000000	0x48	A	G	M	D	K	A	A	N	0001000	0x71	G	V	P	V	V
2	000000001000000000001000	00000000	0x20	V	I	T	P	Q	N	G	A	10000000	0x49	G	M	D	K	A	A	N	G	0001000	0x72	V	P	V	V	N
0	000000000000000000000000	000000000	0x21	I	T	P	Q	N	G	A	G	000000000	0x4A	M	D	K	A	A	N	G	V	0000000	0x73	P	V	V	N	I
0	000000000000000000000000	000000000	0x22	T	P	Q	N	G	A	G	M	000000000	0x4B	D	K	A	A	N	G	V	P	0000000	0x74	V	V	N	I	A
0	000000000000000000000000	000000000	0x23	P	Q	N	G	A	G	M	D	000000000	0x4C	K	A	A	N	G	V	P	V	0000000	0x75	V	N	I	A	T
0	000000000000000000000000	000000000	0x24	Q	N	G	A	G	M	D	K	000000000	0x4D	A	A	N	G	V	P	V	V	0000000	0x76	N	I	A	T	P
0	000000000000000000000000	000000000	0x25	N	G	A	G	M	D	K	A	000000000	0x4E	A	N	G	V	P	V	V	N	0000000	0x77	I	A	T	P	N
1	000000000000000001000000	000000000	0x26	G	A	G	M	D	K	A	A	000000000	0x4F	N	G	V	P	V	V	N	I	0100000	0x78	A	T	P	N	G
1	000000001000000000000000	000000000	0x27	A	G	M	D	K	A	A	N	100000000	0x50	G	V	P	V	V	N	I	A	0000000	0x79	T	P	N	G	A
0	000000000000000000000000	000000000	0x28	G	M	D	K	A	A	N	G	000000000	0x51	V	P	V	V	N	I	A	T	0000000	0x7A	P	N	G	A	G

Reconfiguration of AMchip for sequence analysis



pattern preparation unit

AMchip configuration

Python

GTX and Ethernet communication setup

loading search pattern (sequence 1) in the AMchip

FPGA

loading the Hit pattern(sequence 2)

staring the AMchip calculation

AMChip

readout calculation result

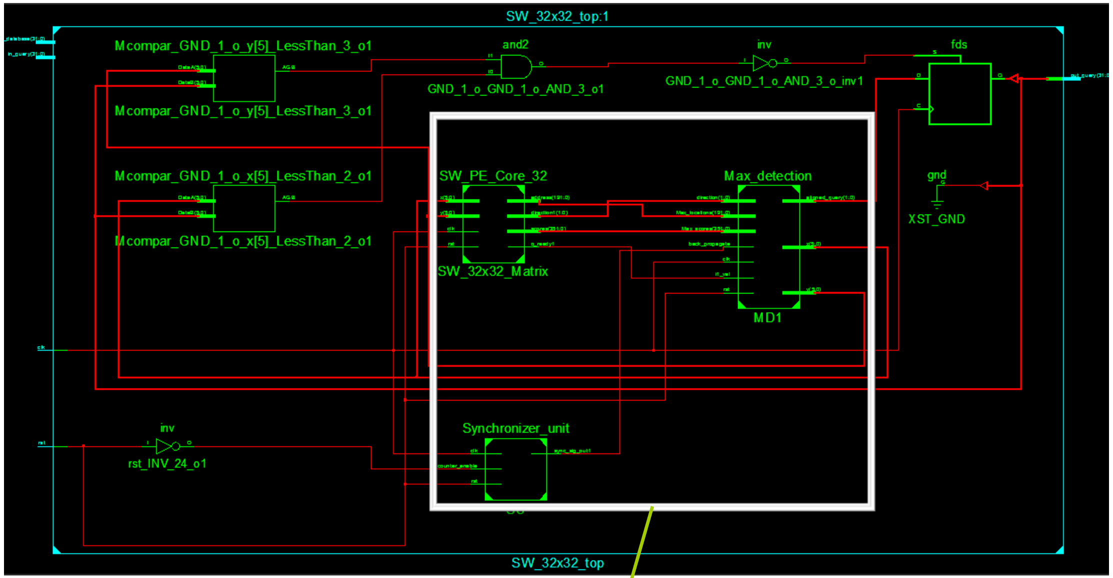
FPGA+AMChip

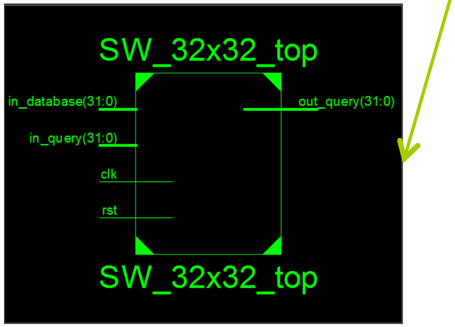
Python

highlighting the detected pattern

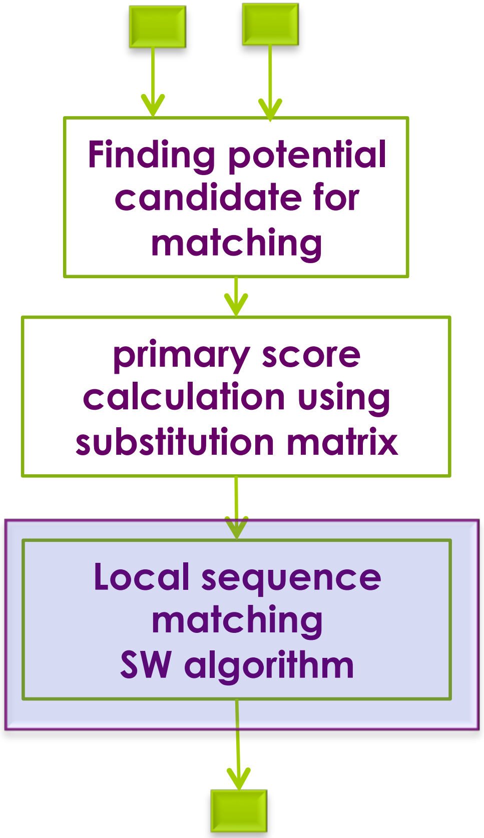
Hardware setup

Post-processing of selected seeds on FPGA


SW_32x32_top:1
Mcompar_GND_1_o_y[5]_LessThan_3_o1
Mcompar_GND_1_o_y[5]_LessThan_3_o1
and2
GND_1_o_GND_1_o_AND_3_o1
inv
GND_1_o_GND_1_o_AND_3_o_inv1
fds
Mcompar_GND_1_o_x[5]_LessThan_2_o1
Mcompar_GND_1_o_x[5]_LessThan_2_o1
SW_PE_Core_32
SW_32x32_Matrix
Max_detection
MD1
gnd
XST_GND
inv
rst_INV_24_o1
Synchronizer_unit
SW_32x32_top



This part of the algorithm is done on an FPGA


Finding potential candidate for matching
primary score calculation using substitution matrix
Local sequence matching
SW algorithm

AMchip Test-bench 1

Sequence alignment on AMchip

IPMC

AMchip Test-bench 2

documentation

IPMC interfacing through JTAG

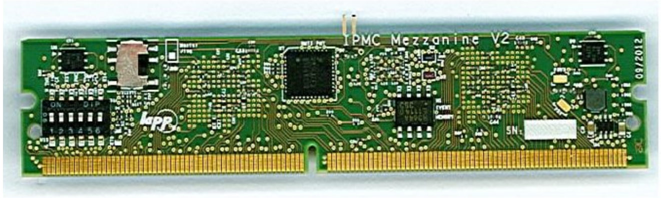
Activities	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16	Mar-16	Apr-16	May-16	Jun-16	Jul-16	Aug-16	Sep-16	Oct-16	Nov-16	Dec-16	Jan-17	Feb-17	Mar-17
AMchip Test-bench I (firmware and software)	🟩	🟩	🟩	🟩	🟩	🟩	🟩																	
Sequence alignment based on AMchip					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩									
IPMC board															🟩	🟩	🟩	🟩						
AMchip Test-bench 2 (firmware and software)																								
presentation and documentstation																								

IPMC JTAG interface

IPMC JTAG interfacing and test

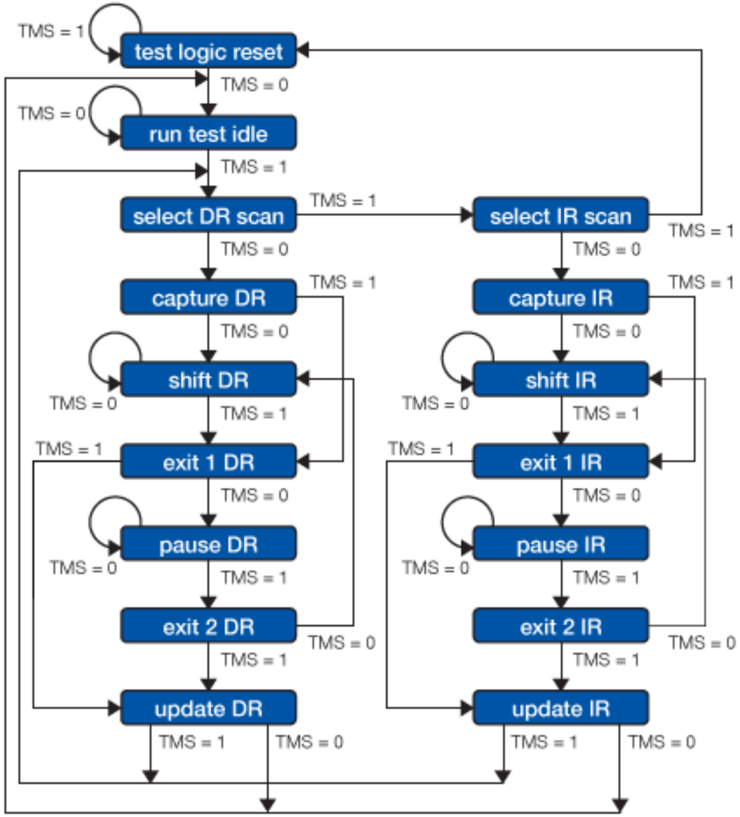
Complementary information: https://indico.cern.ch/event/300897/contributions/1664698/attachments/567236/781334

A request from **LAPP (**our collaborative in Annecy) → the task was completed and submitted by **LPNHE**



LPNHE was involved in JTAG development and now IPMC card is **being used as power management board of CTA and LAr board.**

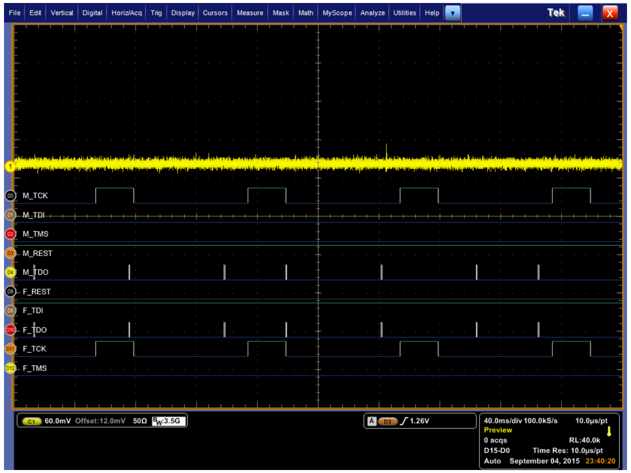
The aim of the JTAG interface testing is to program JTAG IR and DR to perform certain task based on below state machine.

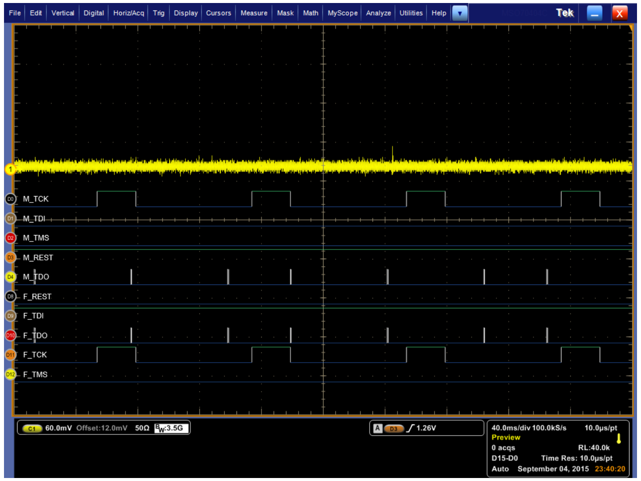


IPMC JTAG interface

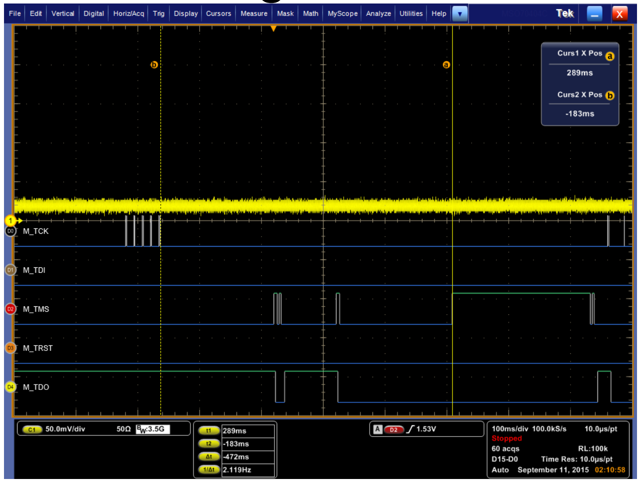
Some result of JTAG test

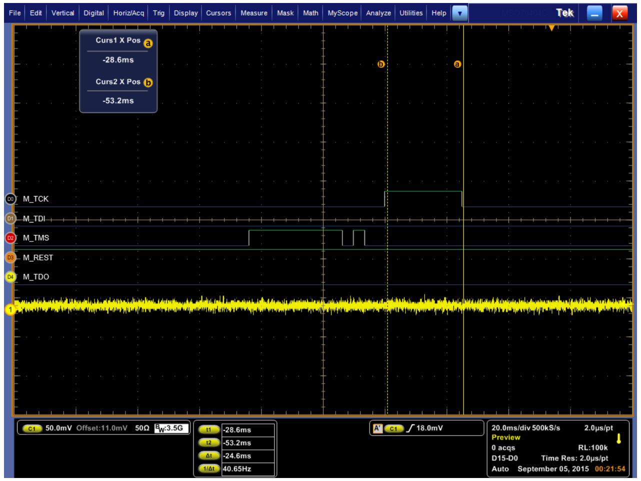
Individual line test


File Edit Vertical Digital Horiz/Acq Trig Display Cursors Measure Mask Math MyScope Analyze Utilities Help
Tek
M_TCK
M_TDI
M_TMS
M_REST
M_TDO
F_REST
F_TDI
F_TDO
F_TCK
F_TMS
C1 60.0mV Offset:12.0mV 50Ω BW:3.5G
A D3 1.26V
40.0ms/div 100.0kS/s 10.0µs/pt
Preview
0 acqs RL:40.0k
D15-D0 Time Res: 10.0µs/pt
Auto September 04, 2015 23:40:20


File Edit Vertical Digital Horiz/Acq Trig Display Cursors Measure Mask Math MyScope Analyze Utilities Help
Tek
M_TCK
M_TDI
M_TMS
M_REST
M_TDO
F_REST
F_TDI
F_TDO
F_TCK
F_TMS
C1 60.0mV Offset:12.0mV 50Ω BW:3.5G
A D3 1.26V
40.0ms/div 100.0kS/s 10.0µs/pt
Preview
0 acqs RL:40.0k
D15-D0 Time Res: 10.0µs/pt
Auto September 04, 2015 23:40:20

Controlling JTAG chain


File Edit Vertical Digital Horiz/Acq Trig Display Cursors Measure Mask Math MyScope Analyze Utilities Help
Tek
Curs1 X Pos a
289ms
Curs2 X Pos b
-183ms
M_TCK
M_TDI
M_TMS
M_TRST
M_TDO
C1 50.0mV/div 50Ω BW:3.5G
t1 289ms
t2 -183ms
Δt -472ms
1/Δt 2.119Hz
A D2 1.53V
100ms/div 100.0kS/s 10.0µs/pt
Stopped
60 acqs RL:100k
D15-D0 Time Res: 10.0µs/pt
Auto September 11, 2015 02:10:58

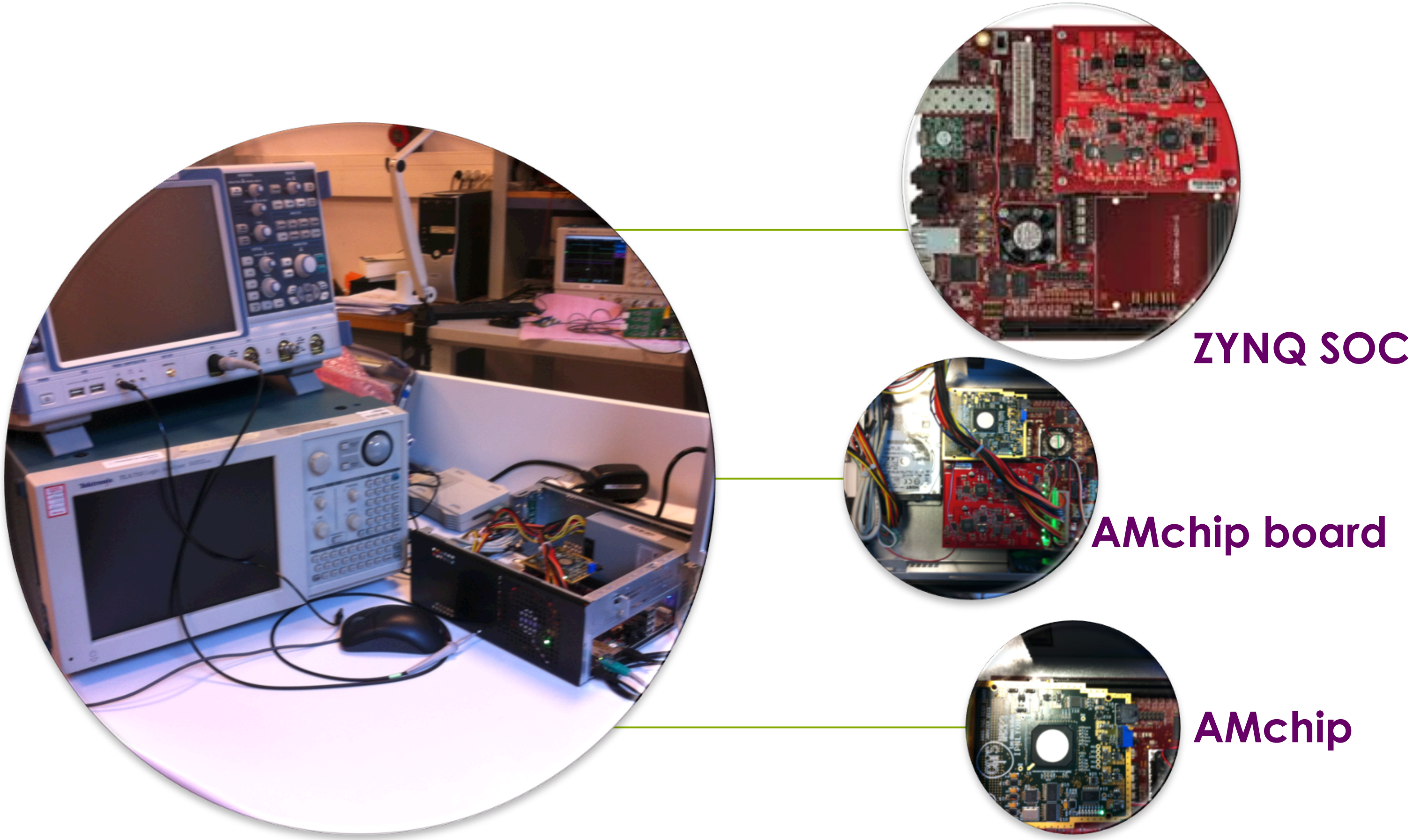

File Edit Vertical Digital Horiz/Acq Trig Display Cursors Measure Mask Math MyScope Analyze Utilities Help
Tek
Curs1 X Pos a
-28.6ms
Curs2 X Pos b
-53.2ms
M_TCK
M_TDI
M_TMS
M_REST
M_TDO
C1 50.0mV Offset:11.0mV 50Ω BW:3.5G
t1 -28.6ms
t2 -53.2ms
Δt -24.6ms
1/Δt 40.65Hz
A C1 18.0mV
20.0ms/div 500kS/s 2.0µs/pt
Preview
0 acqs RL:100k
D15-D0 Time Res: 2.0µs/pt
Auto September 05, 2015 00:21:54

Test-bench 2(firmware and software)

Activities	Apr-15	May-15	Jun-15	Jul-15	Aug-15	Sep-15	Oct-15	Nov-15	Dec-15	Jan-16	Feb-16	Mar-16	Apr-16	May-16	Jun-16	Jul-16	Aug-16	Sep-16	Oct-16	Nov-16	Dec-16	Jan-17	Feb-17	Mar-17
AMchip Test-bench I (firmware and software)	🟩	🟩	🟩	🟩	🟩	🟩	🟩																	
Sequence alignment based on AMchip					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩									
IPMC board															🟩	🟩	🟩	🟩						
AMchip Test-bench 2 (firmware and software)																	🟩	🟩	🟩	🟩	🟩	🟩	🟩	
presentation and documentstation																								

Test-bench 2 (ZYNQ-7 FPGA)

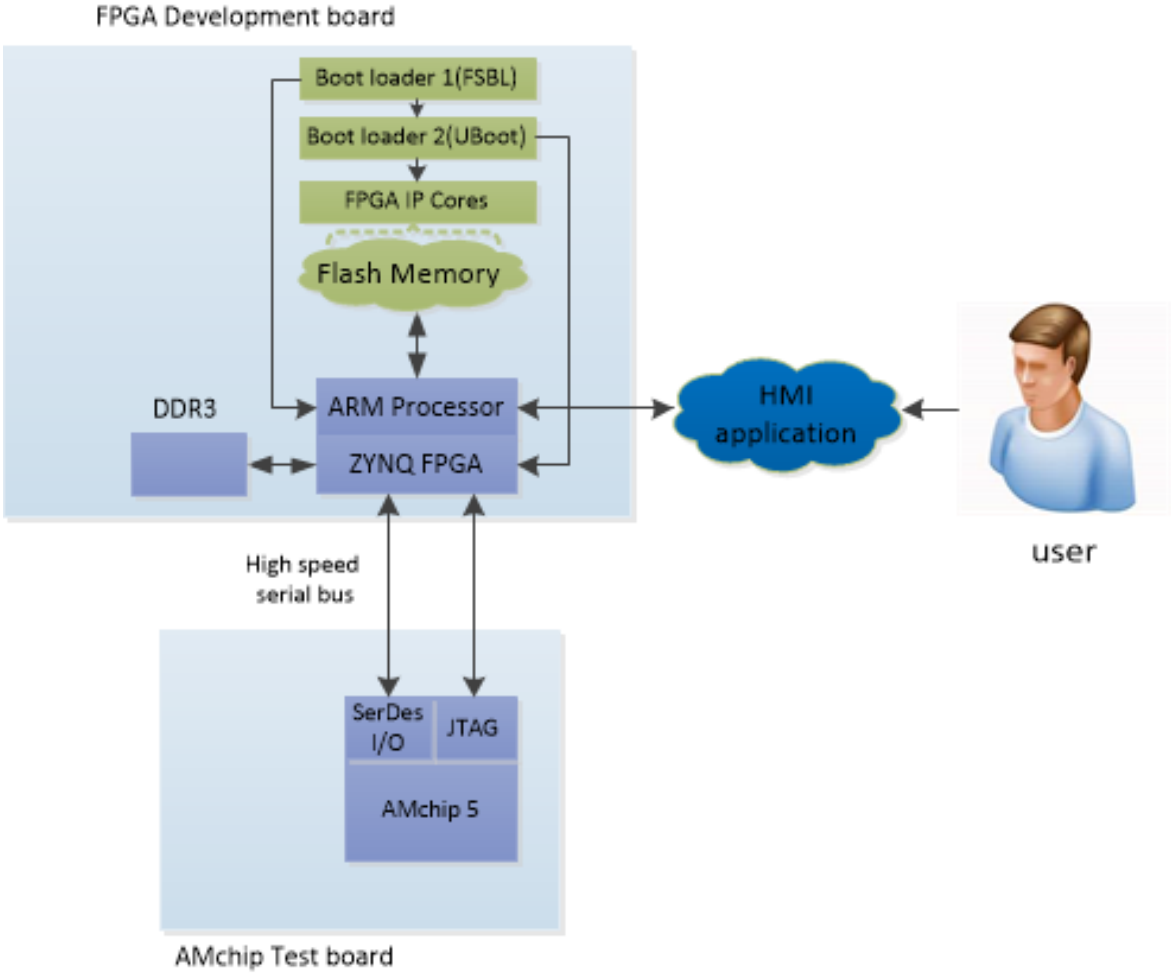
New development platform based on ZYNQ-FPGA and AMchip for different applications



New AMchip test-bench and development platform

Test-bench 2 (ZYNQ-7 FPGA)

New test-bench architecture based on ZYNQ-SOC

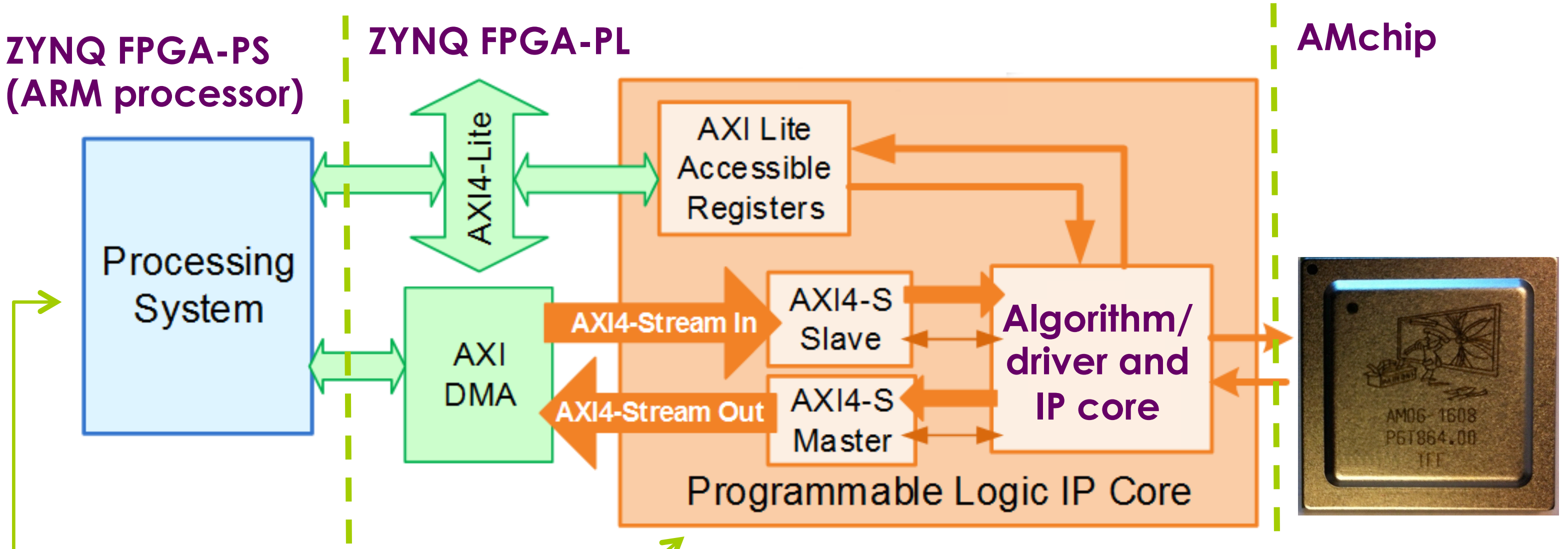


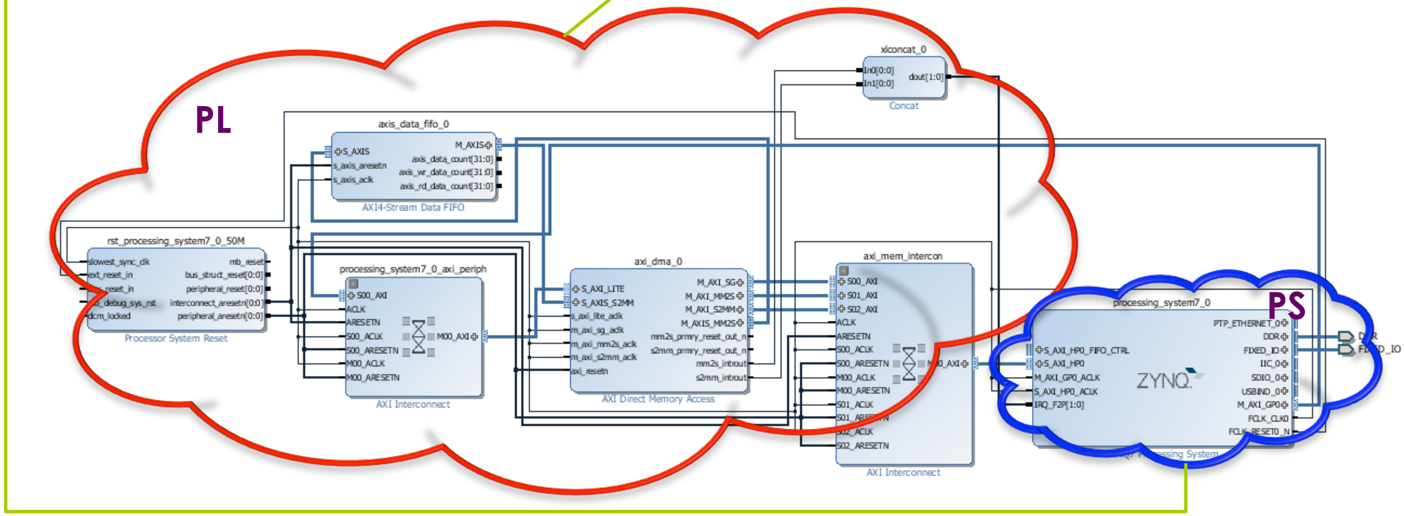
Interfacing from ARM processor to FPGA Embedded Linux

- Component of the embedded Linux
- Interfacing and Communication between ARM and FPGA
- Different library
- Full-chain and gigabyte link
- Some results

Test-bench 2 (ZYNQ-7 FPGA)

Embedded system and Zynq platform

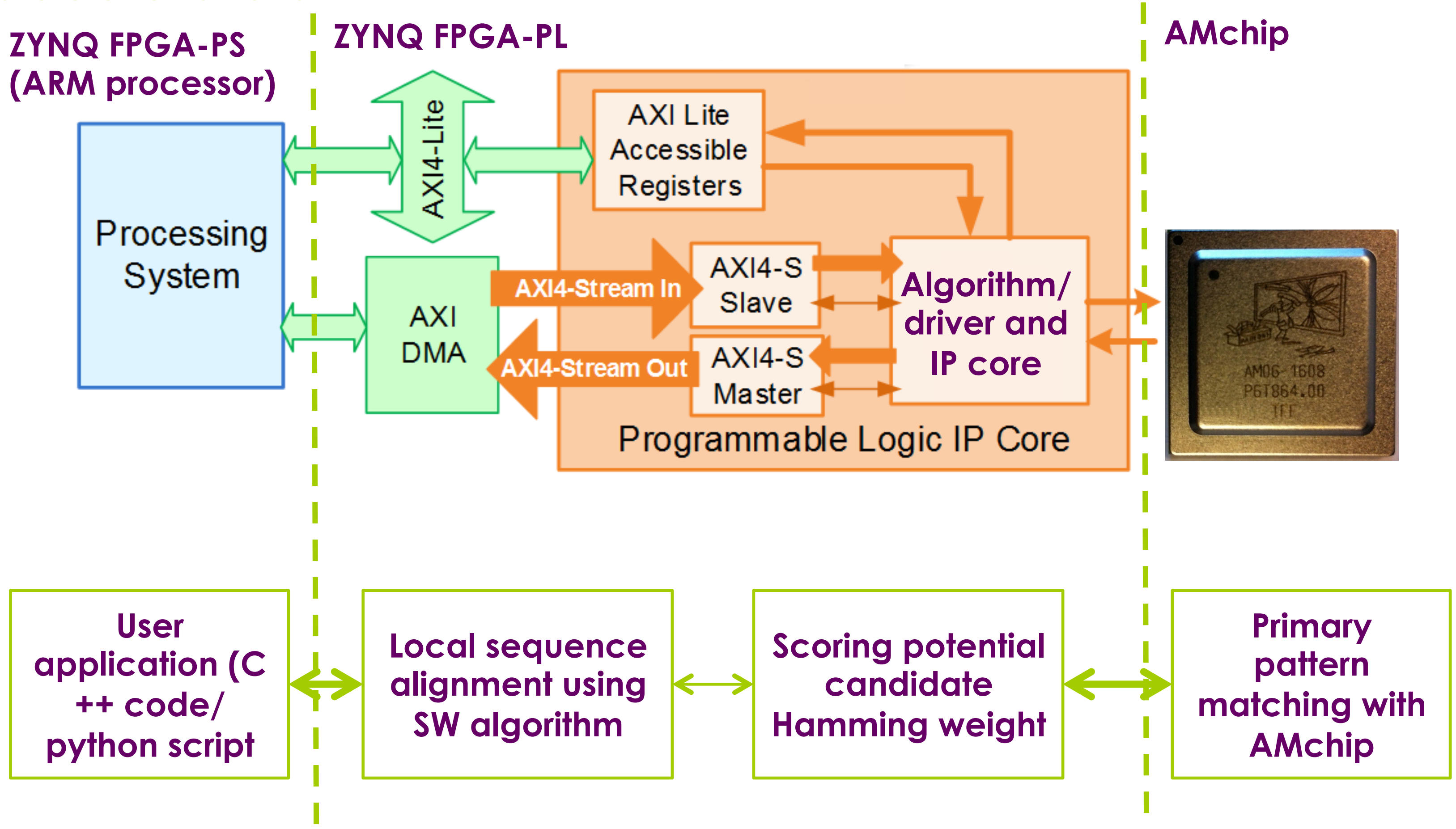




Example of Design integration in the Vivado IP integrator

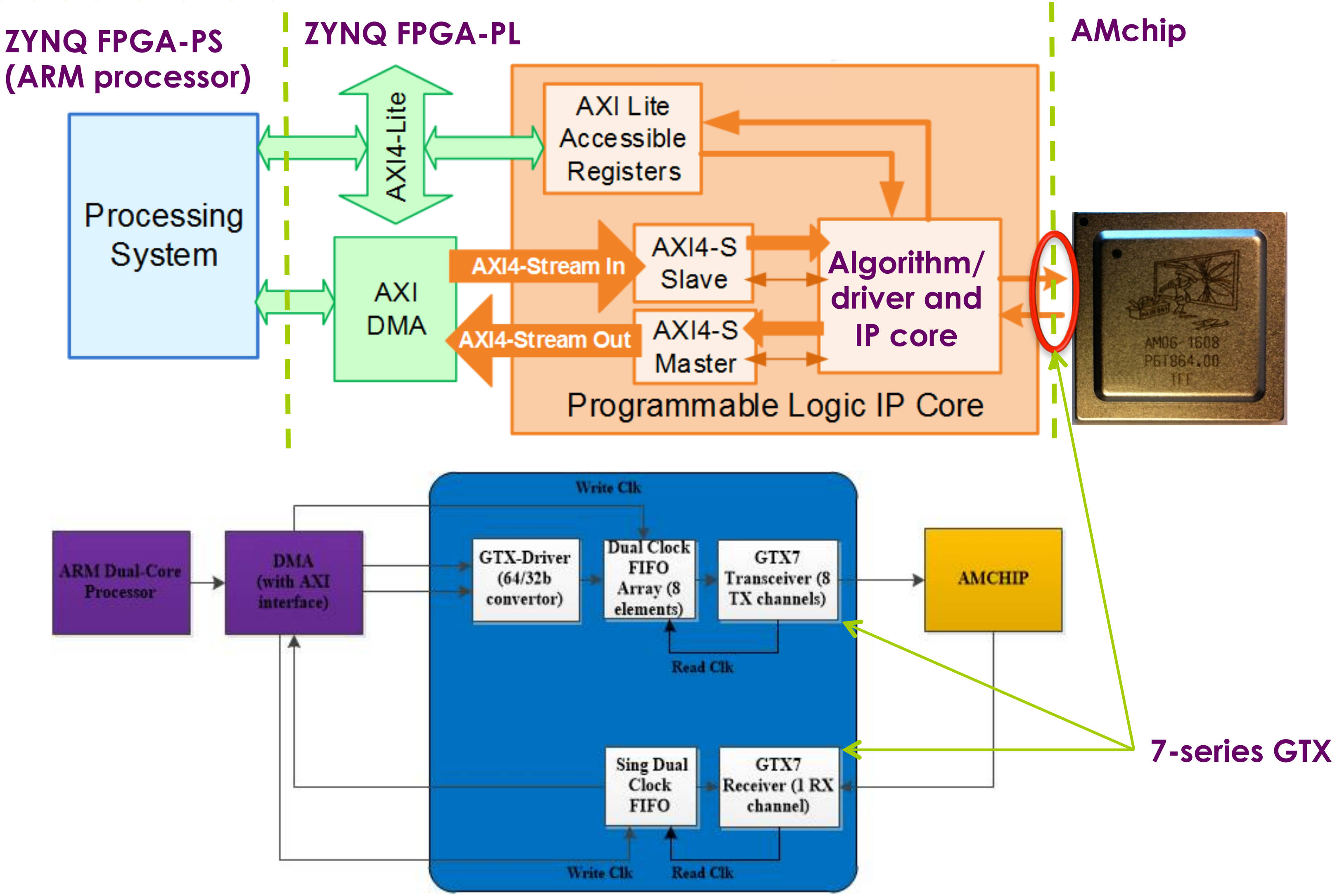
Test-bench 2 (ZYNQ-7 FPGA)

Full-chain high-speed link for data communication and accelerators


ZYNQ FPGA-PS
(ARM processor)
ZYNQ FPGA-PL
AMchip
Processing System
AXI4-Lite
AXI DMA
AXI Lite Accessible Registers
AXI4-Stream In
AXI4-Stream Out
AXI4-S Slave
AXI4-S Master
Algorithm/ driver and IP core
Programmable Logic IP Core
AM06 1608
P6T864.00
User application (C ++ code/ python script
Local sequence alignment using SW algorithm
Scoring potential candidate Hamming weight
Primary pattern matching with AMchip

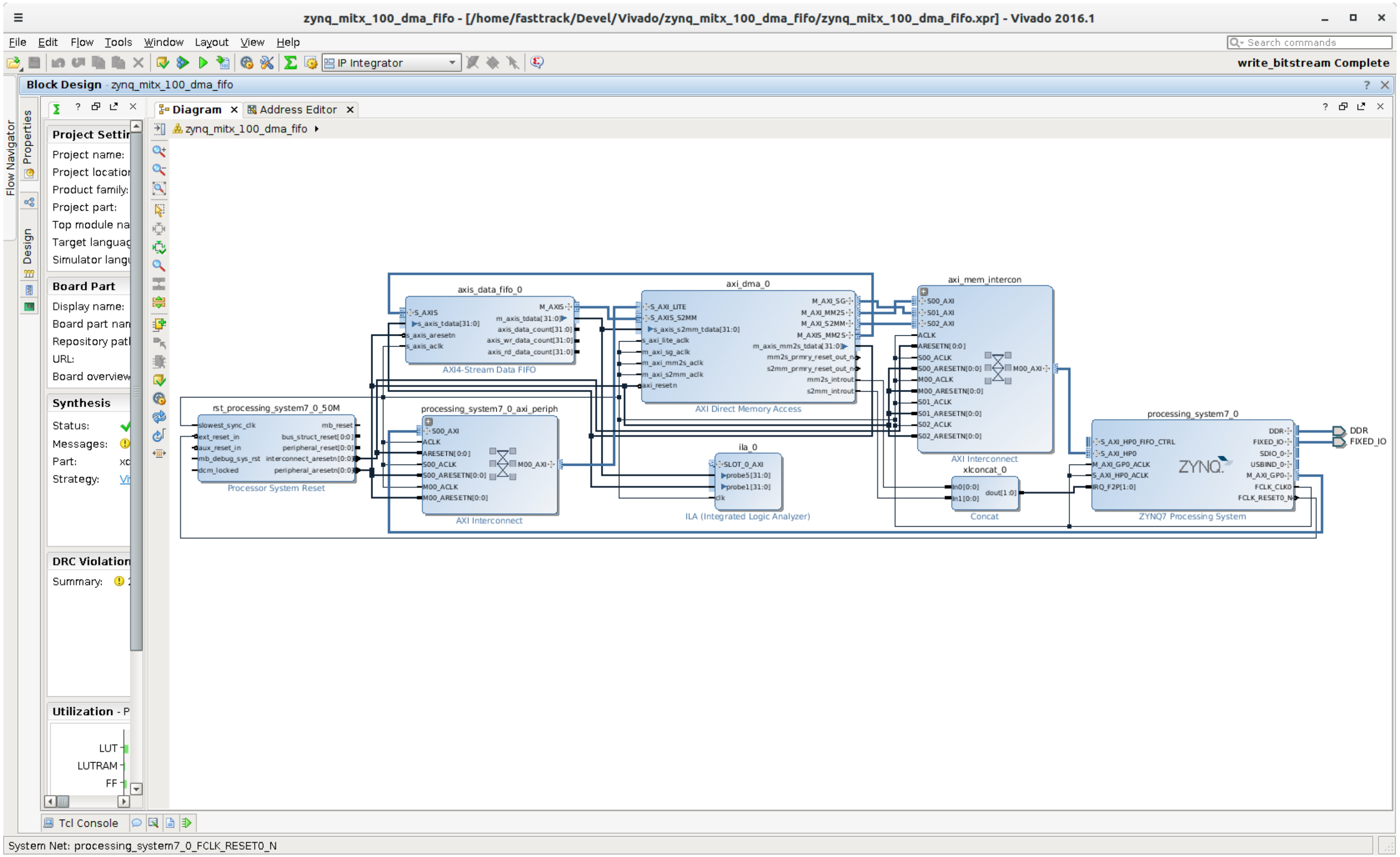
Test-bench 2 (ZYNQ-7 FPGA)

Full-chain high-speed link for data communication and accelerators



Test-bench 2 (ZYNQ-7 FPGA)

Architecture for testing Libgament and Zdma library



zynq_mitx_100_dma_fifo - [/home/fasttrack/Devel/Vivado/zynq_mitx_100_dma_fifo/zynq_mitx_100_dma_fifo.xpr] - Vivado 2016.1

write_bitstream Complete

Block Design - zynq_mitx_100_dma_fifo

Project Settin
Project name:
Project locatio
Product family:
Project part:
Top module na
Target languag
Simulator langu

Board Part
Display name:
Board part nan
Repository pat
URL:
Board overview

Synthesis
Status: ✓
Messages: !
Part: xc
Strategy: Vi

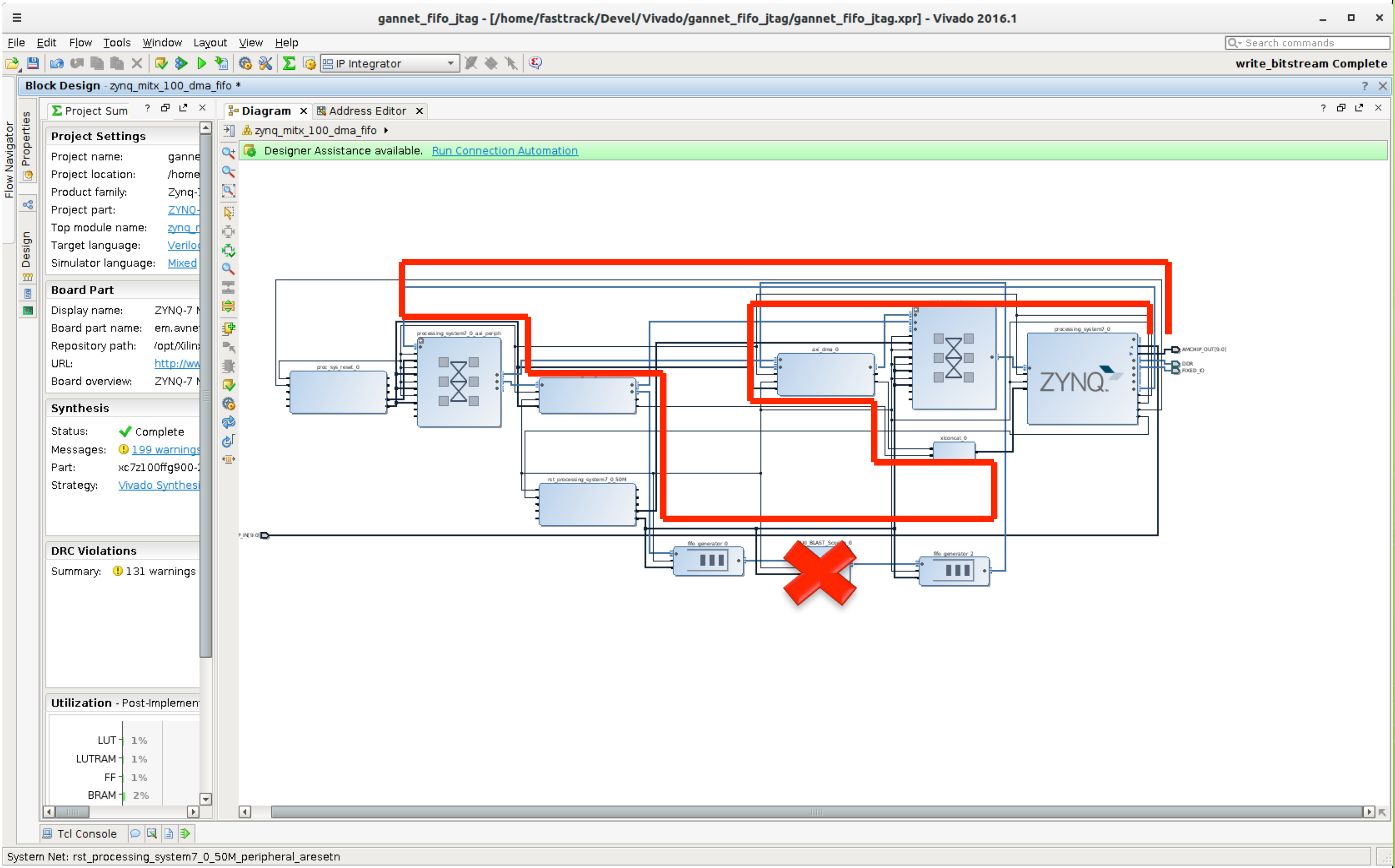
DRC Violation
Summary: ! 2

Utilization - P

System Net: processing_system7_0_FCLK_RESET0_N

Test-bench 2 (ZYNQ-7 FPGA)

Full-chain speed test architecture



gannet_fifo_jtag - [/home/fasttrack/Devel/Vivado/gannet_fifo_jtag/gannet_fifo_jtag.xpr] - Vivado 2016.1

write_bitstream Complete

Block Design - zynq_mitx_100_dma_fifo *

Project Settings

Project name: ganne

Project location: /home

Product family: Zynq-

Project part: ZYNQ-

Top module name: zynq_r

Target language: Verilo

Simulator language: Mixed

Board Part

Display name: ZYNQ-7 M

Board part name: em.avne

Repository path: /opt/Xilin

URL: http://ww

Board overview: ZYNQ-7 M

Synthesis

Status: ✓ Complete

Messages: 199 warnings

Part: xc7z100ffg900-

Strategy: Vivado Synthesi

DRC Violations

Summary: 131 warnings

Utilization - Post-Implemen

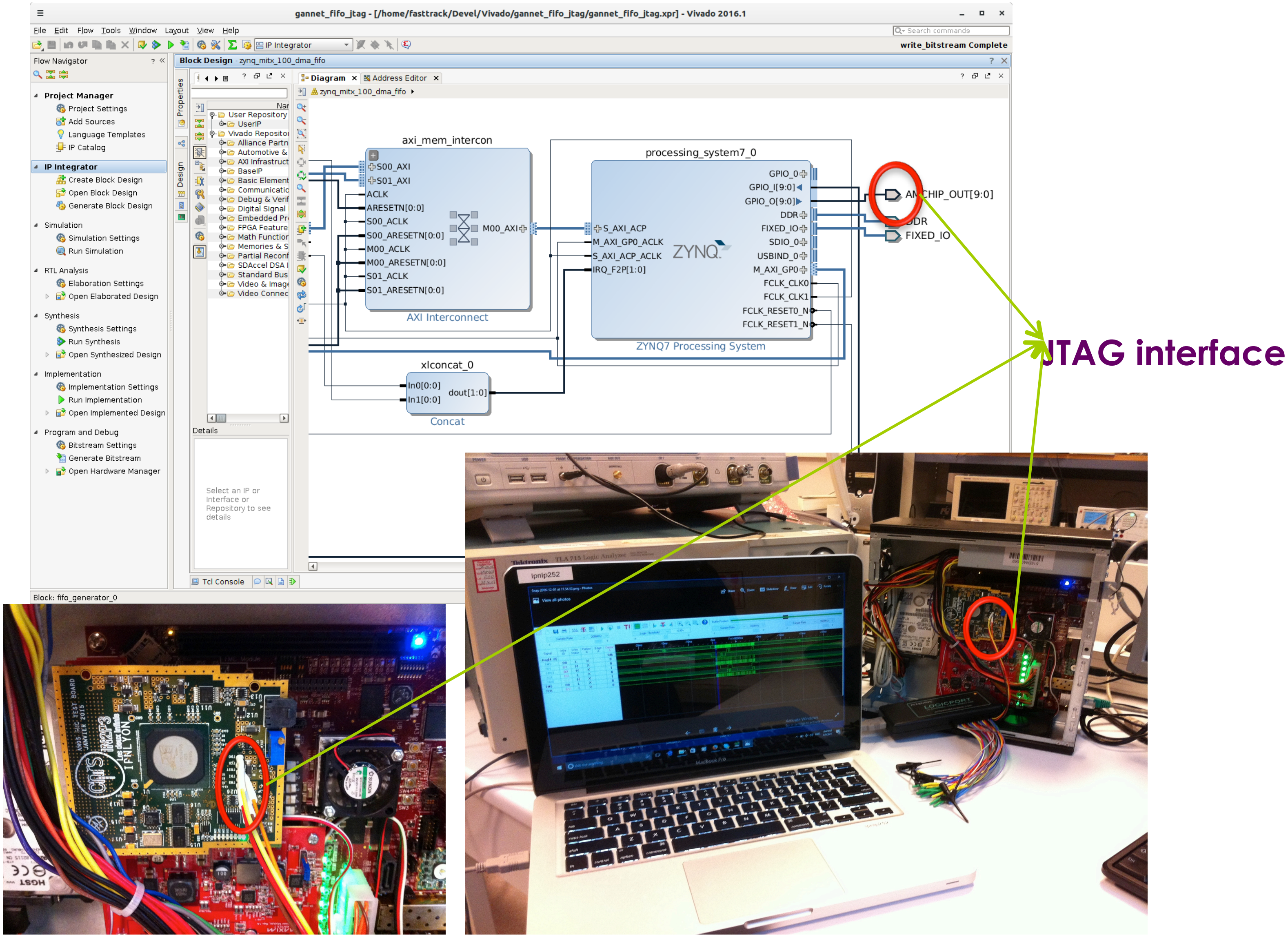
LUT	1%
LUTRAM	1%
FF	1%
BRAM	2%

Designer Assistance available. Run Connection Automation

System Net: rst_processing_system7_0_50M_peripheral_aresetn

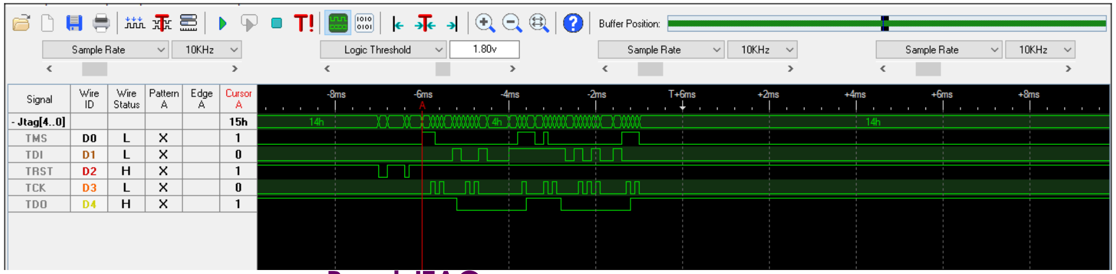
Test-bench 2 (ZYNQ-7 FPGA)

DDR, AMchip and LED connection interface



Test-bench 2 (ZYNQ-7 FPGA)

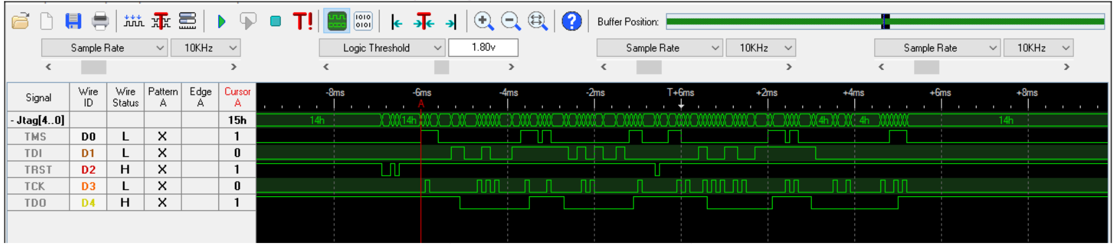
JTAG test



Sample Rate 10KHz — Logic Threshold 1.80v

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
- Jtag[4..0]					15h
TMS	D0	L	X		1
TDI	D1	L	X		0
TRST	D2	H	X		1
TCK	D3	L	X		0
TDO	D4	H	X		1

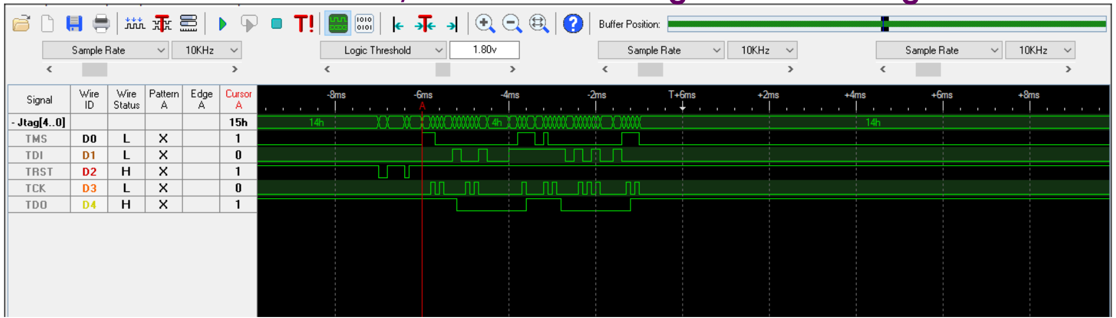
Read JTAG



Sample Rate 10KHz — Logic Threshold 1.80v

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
- Jtag[4..0]					15h
TMS	D0	L	X		1
TDI	D1	L	X		0
TRST	D2	H	X		1
TCK	D3	L	X		0
TDO	D4	H	X		1

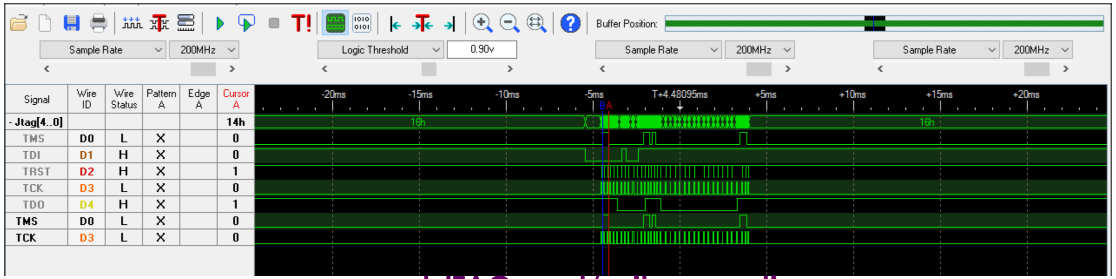
Read/write to SERDES register 0xc9 through JTAG



Sample Rate 10KHz — Logic Threshold 1.80v

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
- Jtag[4..0]					15h
TMS	D0	L	X		1
TDI	D1	L	X		0
TRST	D2	H	X		1
TCK	D3	L	X		0
TDO	D4	H	X		1

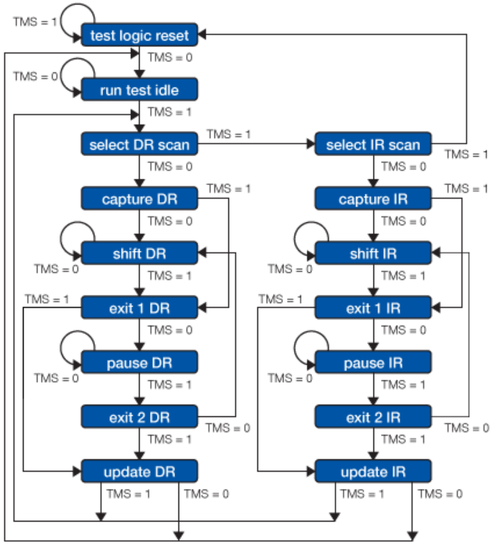
Write to JTAG



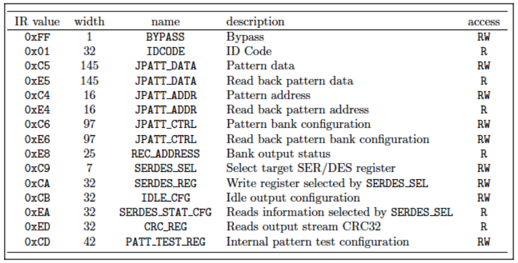
Sample Rate 200MHz — Logic Threshold 0.90v

Signal	Wire ID	Wire Status	Pattern A	Edge A	Cursor A
- Jtag[4..0]					14h
TMS	D0	L	X		0
TDI	D1	H	X		0
TRST	D2	H	X		1
TCK	D3	L	X		0
TDO	D4	H	X		1
TMS	D0	L	X		0
TCK	D3	L	X		0

zoom out JTAG read/write operation



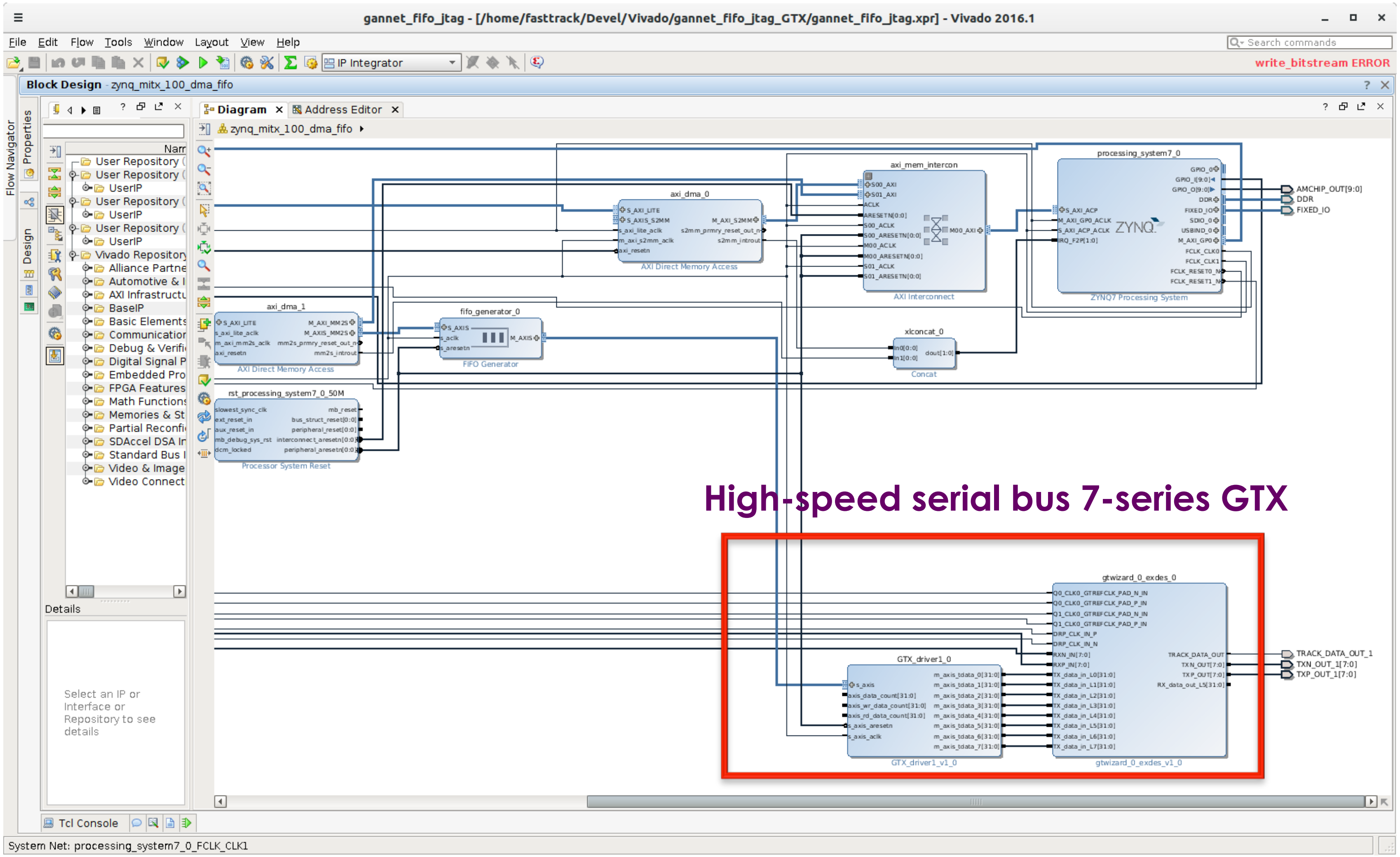
The aim of the JTAG test is to implement following table using above state machine



IR value	width	name	description	access
0xFF	1	BYPASS	Bypass	RW
0x01	32	IDCODE	ID Code	R
0xC5	145	JPATT_DATA	Pattern data	RW
0xE5	145	JPATT_DATA	Read back pattern data	R
0xC4	16	JPATT_ADDR	Pattern address	RW
0xE4	16	JPATT_ADDR	Read back pattern address	R
0xC6	97	JPATT_CTRL	Pattern bank configuration	RW
0xE6	97	JPATT_CTRL	Read back pattern bank configuration	RW
0xE8	25	REC_ADDRESS	Bank output status	R
0xC9	7	SERDES_SEL	Select target SER/DES register	RW
0xCA	32	SERDES_REG	Write register selected by SERDES_SEL	RW
0xCB	32	IDLE_CFG	Idle output configuration	RW
0xEA	32	SERDES_STAT_CFG	Reads information selected by SERDES_SEL	R
0xED	32	CRC_REG	Reads output stream CRC32	R
0xCD	42	PATT_TEST_REG	Internal pattern test configuration	RW

Test-bench 2 (ZYNQ-7 FPGA)

High-speed GTX part I



gannet_fifo_jtag - [/home/fasttrack/Devel/Vivado/gannet_fifo_jtag_GTX/gannet_fifo_jtag.xpr] - Vivado 2016.1

write_bitstream ERROR

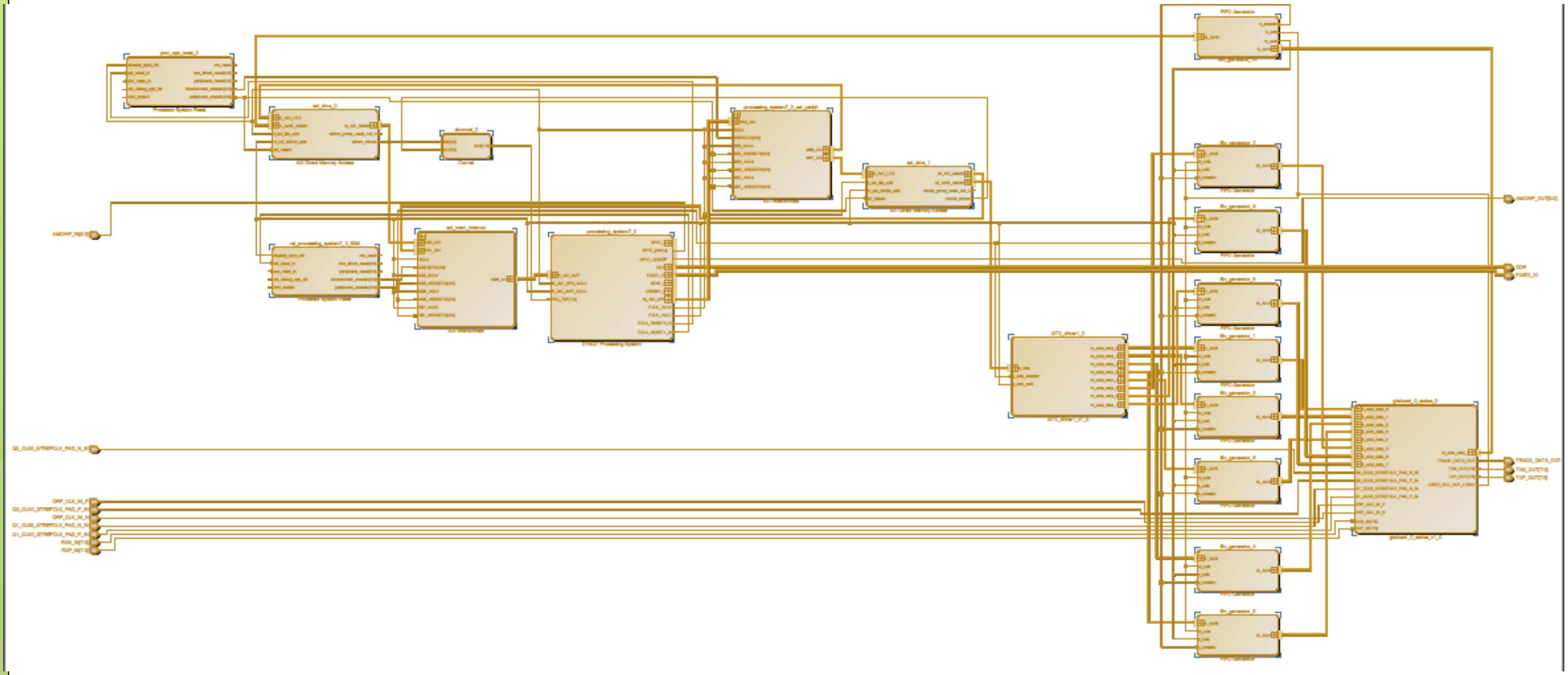
Block Design - zynq_mitx_100_dma_fifo

System Net: processing_system7_0_FCLK_CLK1

High-speed serial bus 7-series GTX

Test-bench 2 (ZYNQ-7 FPGA)

Full chain link including GTX



Summary and documentation

- A wiki has already been setup
- Further help is needed to document the developments and distribute
- another technical meeting will be set soon to explain above activities in more detail
- Integrating AMchip in more applications
- Using current platform and sharing knowledge with other researcher