



# Embedded Software Development for MPSoC: Preliminary HOPES Experience

June 24, 2008

Soonhoi Ha  
Seoul National University



1



## Contents



1. Introduction: HOPES Design Flow



2. Common Intermediate Code (CIC) and CIC Translation



3. Preliminary Experiments



4. Discussion and Conclusion



2

HOPES project, SNU

# Problem Statement

## Target: MPSoC (Embedded Systems) with high degree of parallelism

- Scalability
- Heterogeneous processors with diverse communication architecture
- Power-constrained system

## Problem: parallel programming for MP Systems

- Parallelism extraction (multiple use case, multi-tasking apps.)
  - Functional parallelism, data-parallelism, temporal-parallelism
- Partitioning and mapping
- Parallel code generation: parallel programming is not easy
- Performance estimation and verification
- Design space exploration

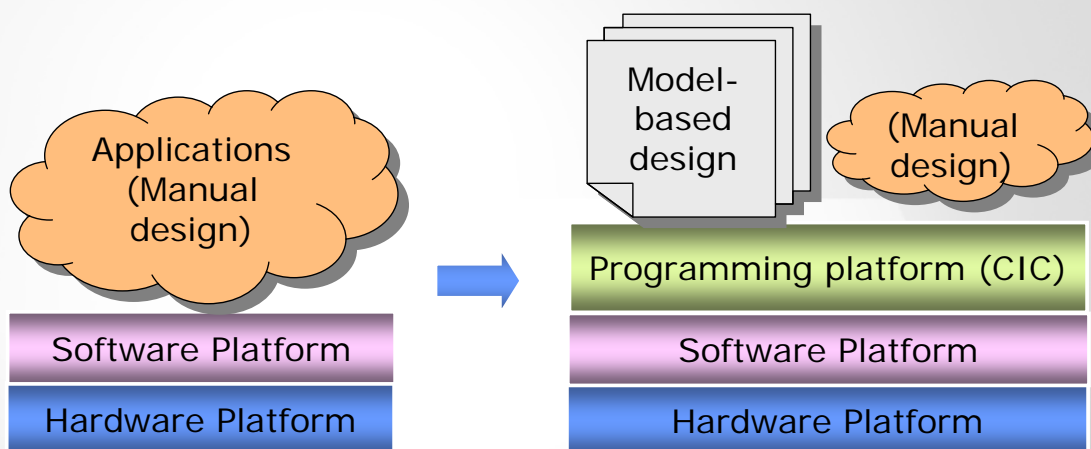


Need of sound (scalable and robust) methodology

3

HOPES project, SNU

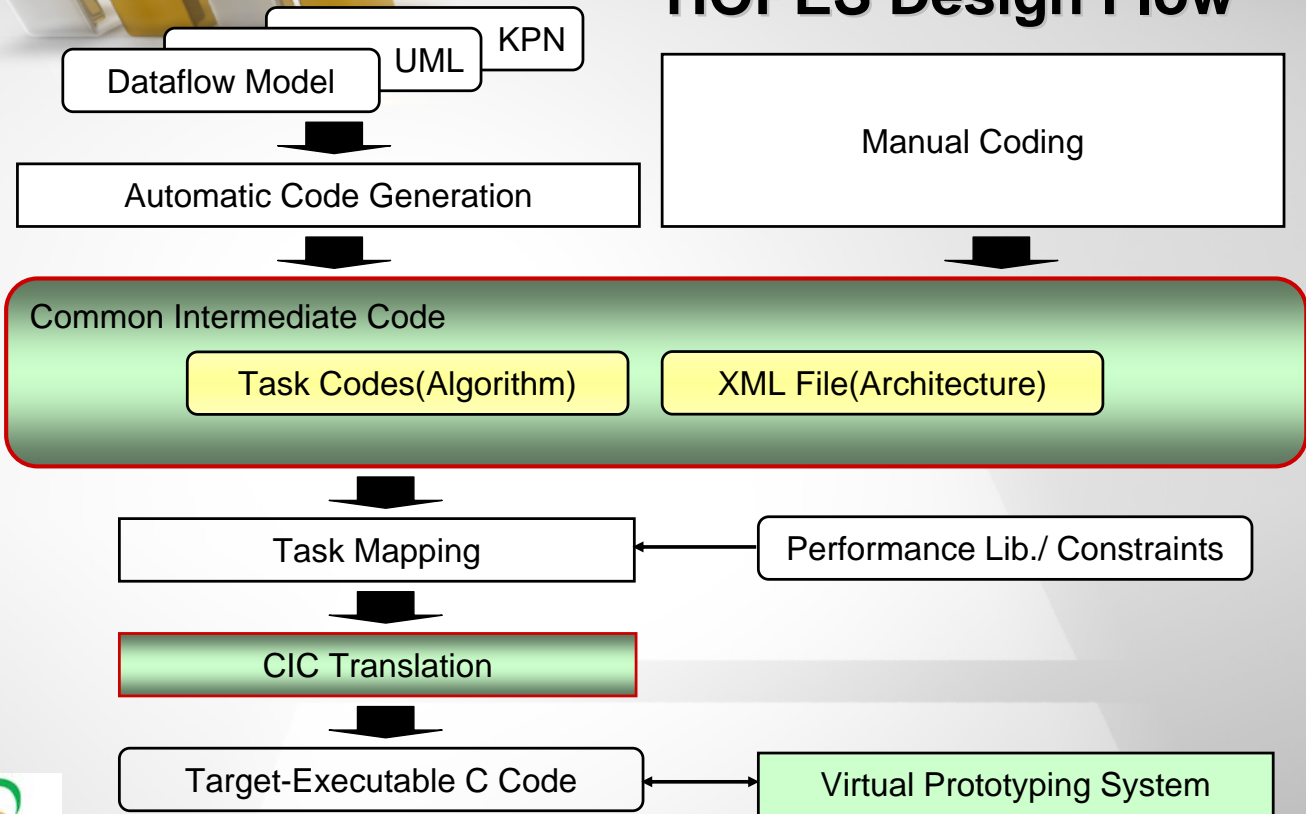
# Basic Idea



4

HOPES project, SNU

# HOPES Design Flow



5

HOPES project, SNU

## Design Steps and Techniques

- ✦ **Algorithm development / Behavior specification**
  - (Option 1) PeaCE modeling: FSM + Dataflow tasks  
Automatic clustering into CIC tasks
  - (Option 2) CIC task specification
- ✦ **Functional Simulation**
  - Multithreaded code generation in the host machine
- ✦ **Mapping to the processing elements**
  - Automatic mapping considering functional/data/temporal parallelism.
  - Manual mapping
- ✦ **(Architectural design space exploration)**
- ✦ **CIC translation to generate target C codes**
- ✦ **Virtual prototyping / system verification**



6

HOPES project, SNU

1. Introduction: HOPES Design Flow

2. Common Intermediate Code (CIC) and CIC Translation

3. Preliminary Experiments

4. Discussion and Conclusion



## CIC (Common Intermediate Code)

.CIC

### CIC (tasks)

Interfaces:

\*\_init()  
\*\_go()  
\*\_wrapup()

Depends on:

Generic APIs  
(including comm. APIs)

Additional information:

OpenMP pragmas  
Hardware pragmas

### CIC (architecture) - .xml files

Hardware information:

Processor information  
H/W accelerator info.  
OS, Memory maps

Constraints:

Timing requirement  
Memory constraint  
Power consumption

Structure:

Task structures  
Channel & Task mapping



# CIC Code Generation

- Separate CIC task codes for partitioned tasks
  - Explicit functional parallelism → task\_name.cic
- openMP programming for data parallelism
- Generic API for platform independent programming

```
void h263decoder_go (void) {  
    ...  
    l = MQ_RECEIVE("mq0", (char *) (ld_106->rdbfr), 2048);  
    ...  
    #pragma omp parallel for  
    for(i=0; i<99; i++) {  
        //thread_main()  
        ....  
    }  
    // display the decode frame  
    dither(frame);  
}
```



# DivX player CIC xml (Structure)

```
<structure>  
  <mode name="default">  
    <task name="AviReaderIO">  
      <subtask name="arm926ej-s0">  
        <procMap>0</procMap>  
        <fileName>AviReaderIO_arm926ej_s0.cic</fileName>  
      </subtask>  
    </task>  
    <task name="H263FRDivxI3">  
      <subtask name="arm926ej-s0">  
        <procMap>0</procMap>  
        <fileName>H263FRDivxI3_arm926ej_s0.cic</fileName>  
      </subtask>  
    </task>  
    <task name="MADStreamI5">  
      <subtask name="arm926ej-s0">  
        <procMap>0</procMap>  
        <fileName>MADStreamI5_arm926ej_s0.cic</fileName>  
      </subtask>  
    </task>  
  </mode>
```

```
<queue>  
  <name>mq0</name>  
  <src>AviReaderIO</src>  
  <dst>H263FRDivxI3</dst>  
  <size>30000</size>  
</queue>  
<queue>  
  <name>mq1</name>  
  <src>AviReaderIO</src>  
  <dst>MADStreamI5</dst>  
  <size>30000</size>  
</queue>  
</structure>  
</CIC_XML>
```





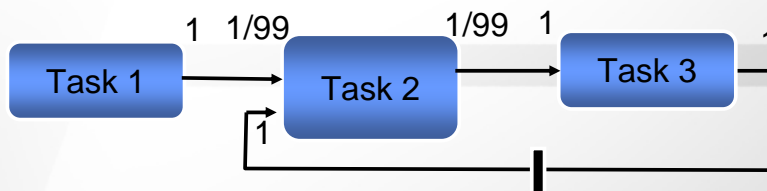
# Task Model in CIC

## Task execution semantics

- Process networks: blocking read, non-blocking write if queue space is available
- Possibility of data parallel execution is specified
  - If there is no internal state inside a task, it can be executed in parallel

## Two types of channels

- FIFO channel of finite size FIFO queue (default)
  - *MQSend(data), MQRead(data)*
- Array channel of finite size array
  - *ACSend(index,data), ACRead(index, data)*



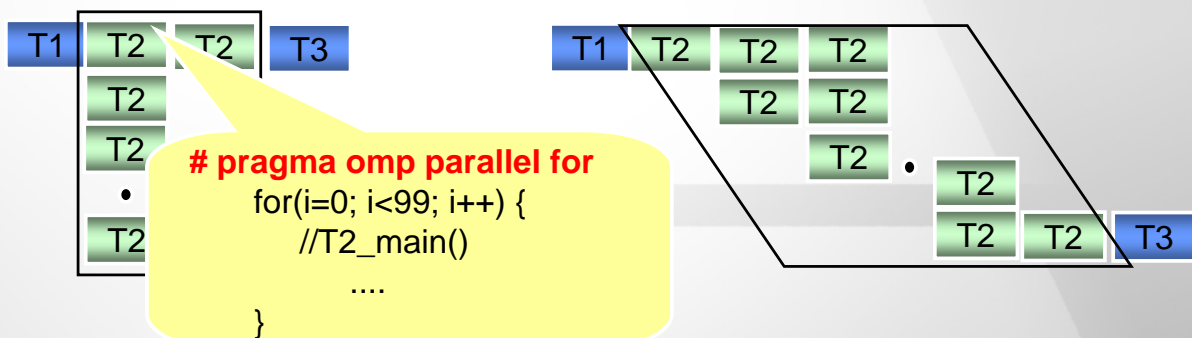
# Data Parallel Execution

## Default parallelism

- (example) Macroblock decoding of H263 decoder
- Specification method: openMP pragma

## Wavefront parallelism

- (example) Macroblock analysis of H264 encoder
- Specification method: CIC definition of wavefront vector



# CIC Translation

## ❏ CIC to Multi-thread codes for functional simulation

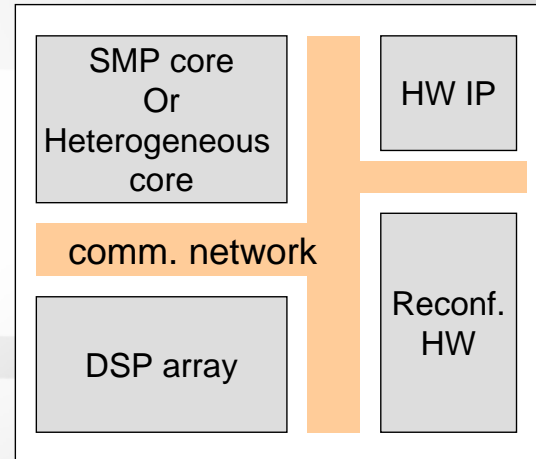
- Generated codes are run on a host machine

## ❏ CIC to target C codes

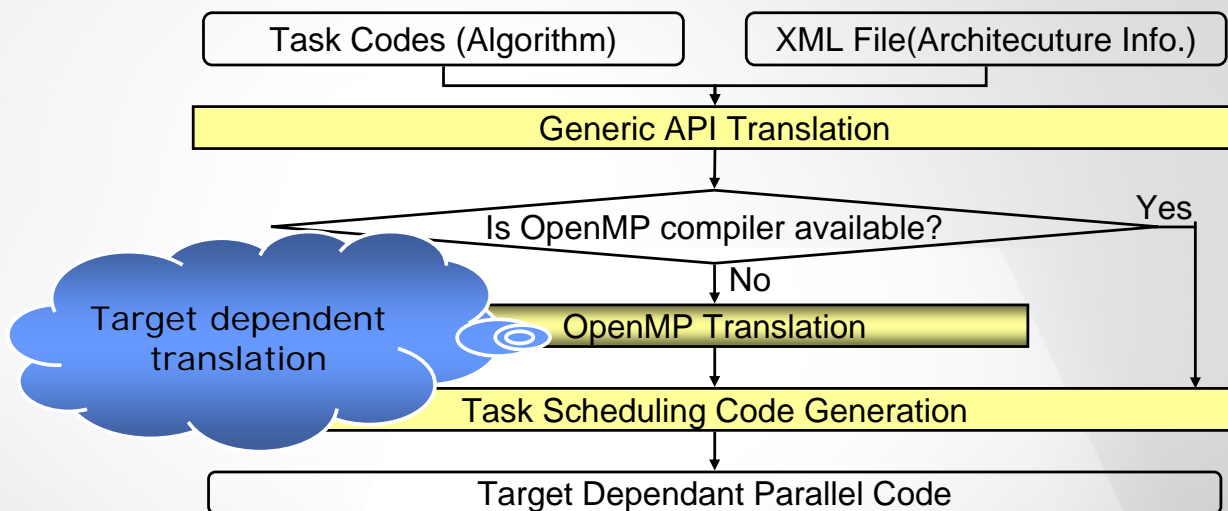
- For virtual prototyping
- For MPCore
- For Cell processor

[planned]

- DSP array
- Reconf. hardware



# CIC Translator









# CIC Mapping (Manual Mapping)

(Manual) Mapping

The screenshot shows the 'schematic.H263Codec' application with the 'Mapping' tab selected. It displays processor configuration, target options, and a task mapping table.

Sub Task	arm926ej_s[0]	arm926ej_s[1]	arm926ej_s[2]	arm926ej_s[3]	arm926ej_s[4]	arm926ej_s[5]
Task: H263FREncIO (4 tasks)						
arm926ej_s_0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
arm926ej_s_1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
arm926ej_s_2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
arm926ej_s_3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Task: H263Dec2I3 (1 tasks)						
arm926ej_s_1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Task: DisplayI9 (1 tasks)						
arm926ej_s_1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Mapping CIC tasks into the target processing elements

# CIC Translation

CIC Translation

Target C codes

```

1
2 // for CIC port api
3 #include "port.h"
4
5 /* User: jwkim
6 Date: Mon Apr 7 16:00:33 2008
7 Target: CIC-XML_Task-Model
8 Universe: H263FREncIO_arm926ej_s_3 */
9
10 /* Define macro for prototyping functions on ANSI & non-ANSI
11 compilers */
12 #ifndef ARGS
13 #defi
14 #else
15 #defi
16 #endi
17 #endif
18
19
20 /* Define constants TRUE and FALSE for portability */
21 #ifndef TRUE
22 #define TRUE 1
23 #endif
24 #ifndef FALSE
25 #define FALSE 0
26 #endif
27
28 /* Define a complex data type if one has not been defined */
29 #if !defined(COMPLEX_DATA)
30 #define COMPLEX_DATA 1
31 typedef struct complex_data { double real; double imag; } com
plex;

```

H.263 encoder

```

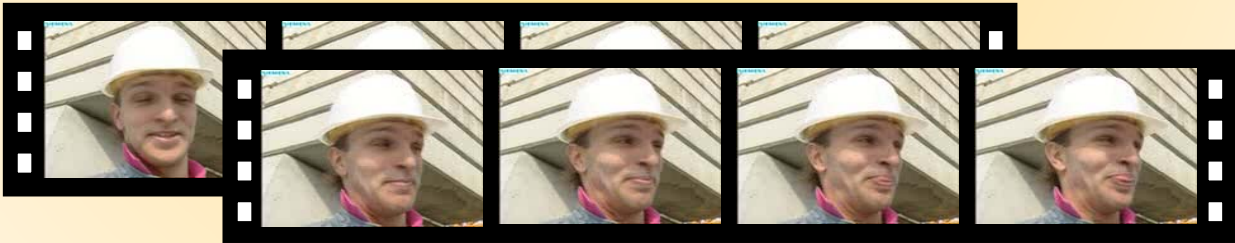
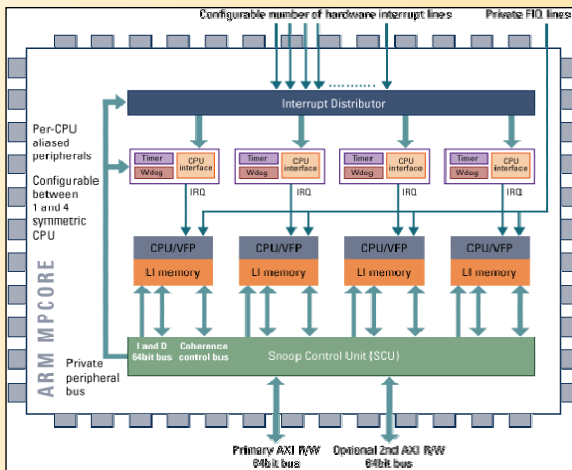
1436
1437 {
1438 (output_792)=((int)(0,00000000000000000000e+00));
1439
1440 /* -- start of #pragma omp parallel -- */
1441 {
1442   parade_thread_shared_H263Dec2I3_arm926ej_s_1_go_0_par_t_
shared;
1443 /* initialize parade_thread_shared_H263Dec2I3_arm926ej_s_
1444   1_go_0 */
1445 /* create threads and do computation */
1446   parallelize(parade_thread_main_H263Dec2I3_arm926ej_s_1_go
_0, &_par_t_shared);
1447 /* -- end of #pragma omp parallel -- */
1448
1449 {
1450   auto int numMacroBlocks=((176)*(144))/((16)*(16));
1451   auto int k;
1452   auto int i;
1453   {
1454     for (i=0; i<numMacroBlocks; i++)
1455     {
1456       (dy)=*((dy_797)+((99)-(offset)));
1457       if (((*(mode_798)+(((90)-(offset))))==(99))>((99)-(((90)-(offset)
)))+(99)-((99))):(((90)-(offset))+((99)))=(0)l|((*(mode_798)+((
90)-(offset))+((99))>((99)))?(((90)-(offset))+((99))-((99))):(((90)
)-(offset))+((99)))=(1))){
1458         reconBlockIntoImage(x_curr,y_curr,dx,dy,((unsigned char *)((outp
utY_780)+(prevY_822)->data)),((output_761)+((395)-((4)*(offset)
))+((3)))->data,((unsigned char *)((outputY_780)+(outputY_825)->
data)),176);
1459         reconBlockIntoImage(x_curr+(0),y_curr,dx,dy,((unsigned char *)((
outputY_780)+(prevY_822)->data)),((output_761)+((395)-((4)*(offse
t))+((2)))->data,((unsigned char *)((outputY_780)+(outputY_8
25)->data)),176);

```

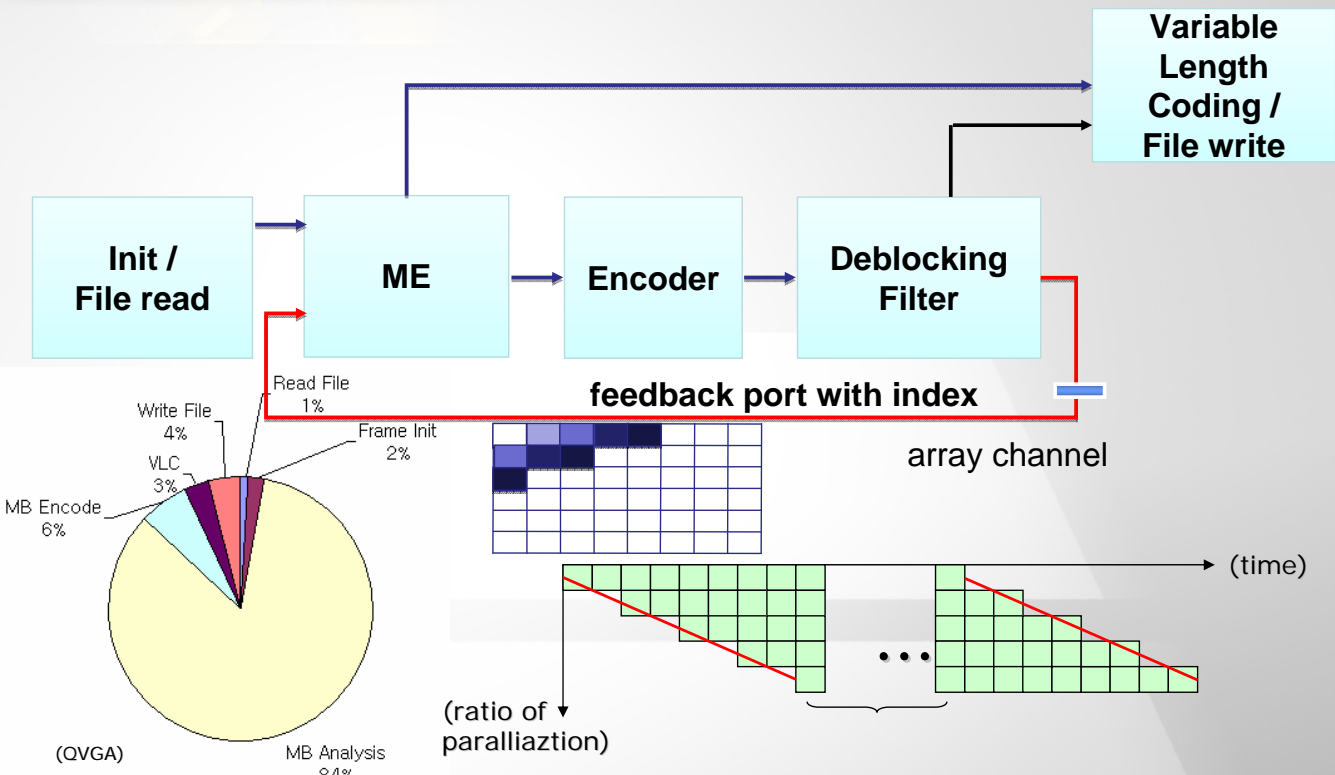
H.263 decoder  
(parallelized code)

# Prototyping

# Prototyping

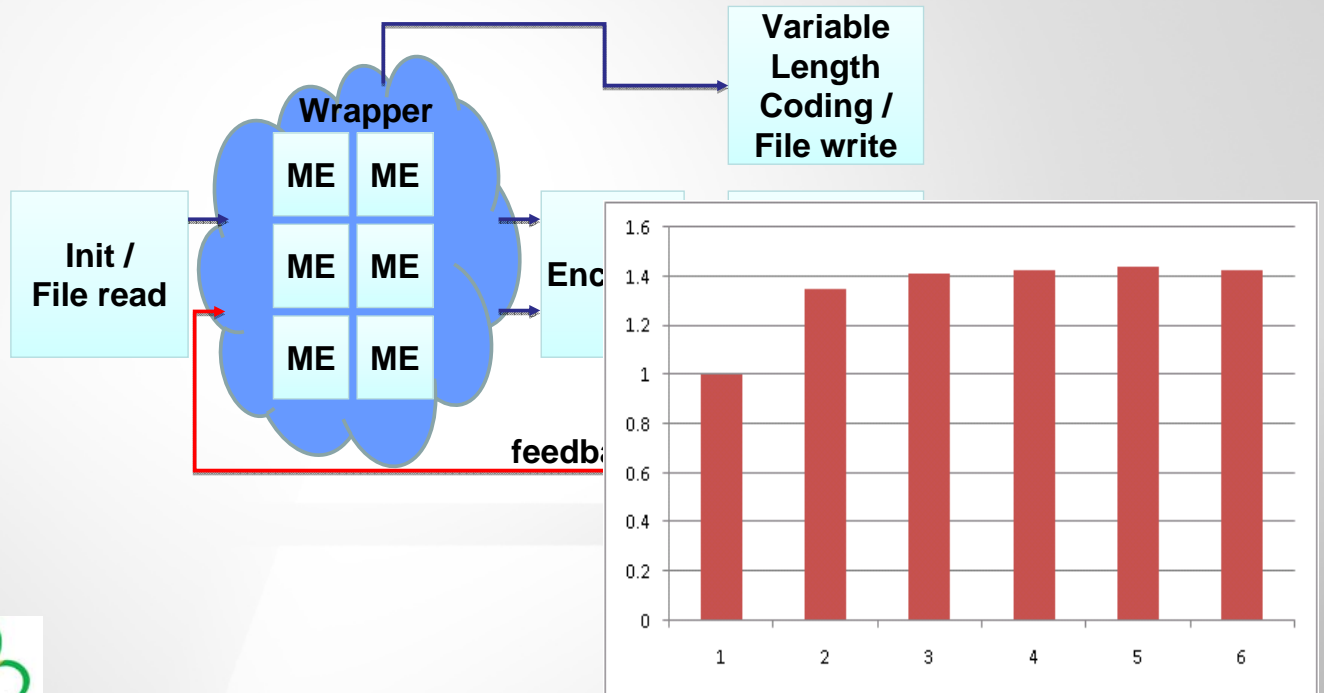


# Manual Design: X264 Encoder on CELL



# X264 Encoder on CELL(2)

- ❏ CIC translator synthesizes wrapper automatically



# Contents

1. Introduction: HOPES Design Flow
2. Task Model in Common Intermediate Code (CIC)
3. Preliminary Experiments
4. Discussion





- ✉ **Currently two ways of CIC generation are supported**
  - PeaCE models to CIC: FSM model to CIC is under development
  - Manual specification of CIC tasks
  - (Simulink Model to CIC, UML to CIC might be interesting)
- ✉ **CIC model refinement**
  - X264 specification uncovers need of extensions.
  - More examples will do the same
- ✉ **Automatic mapping**
  - Currently considers functional and data parallelism only
  - Will consider pipelining too
- ✉ **Virtual prototyping**
  - Will be developed as a TLM simulator that models the communication architecture accurately.
  - Debugging and monitoring features will be added
  - Fast simulator based on virtual synchronization technique is under development



- ✉ **HOPES is a newly launched project to make a embedded software development environment for MPSoCs**
  - Support of diverse models
  - Target independent environment + target specific libraries
  - Integration of software modules at various stages
  - <http://peace.snu.ac.kr/hopes>

