

HABS Developer's Manual (v0.1)

Kiyeon Lee, Charmgil Hong, and Sangyeun Cho
Dept. of Computer Science
University of Pittsburgh
lee, chh50, cho@cs.pitt.edu

Aug 3, 2008

Contents

1 Overview	2
2 Compilation and Usage	2
3 Templates	2
4 HABS XML parser	3
4.1 HABSXlator	3
4.2 HABSXlatorUtils	3
5 HABS translator	3
5.1 System	4
5.2 Processor Chip	4
5.3 Memory Chip	5
5.4 Chip Connectivity	6
5.5 Core Cluster	6
5.6 Hardware IP	7
5.7 On-chip Memory	7
5.8 Processor Core	8
5.9 In-cluster Memory	9
5.10 On-chip Connectivity	10
5.11 In-core Memory	11

1 Overview

This report provides a guide for developers using multicore hardware abstraction (HABS) tool set. HABS tool set is a hardware abstraction model that provides a sufficient amount of hardware information in an abstract manner for the SAIT Component-Based Software Development Project (“SDP”). HABS tool set consists of two parts. One is the parser to parse the machine description written in xml file. The other is the translator to translate the hardware information from the xml file to the memory using the parsed information. The tool set can not only be used to obtain the static information such as the name of the system or type of the processor core, but it can also estimate the hardware information such as the total capacities of memory in the system or the performance of a processor core measured in MIPS given the clock frequency provided by the user.

2 Compilation and Usage

A makefile is included for the user to ease the compilation process. The target is specified as `habs` in the makefile. Once `habs` is produced, the user can use the top-level xml file name, for example, `system.xml`, as a first argument. i.e. `make habs` and `habs system.xml`. The tool set was compiled and tested with `g++ 4.2.1` on a 64-bit Linux machine.

3 Templates

HABS tool set includes a parsing tool and 11 hardware templates: template for system-level description, template for processor chip, template for memory chip, template for chip-level connectivity, template for processor core cluster, template for hardware IP, template for on-chip memory, template for processor core description, template for in-cluster memory, template for on-chip connectivity, and template for in-core memory. Each template is implemented as a class using C++.

- `HABSXlator.cpp`, `HABSXlator.h`
 - parses a given xml file, and reads the described hardware information or counts the number of specific components used in the hardware description xml file
- `HABSXlatorUtils.cpp`, `HABSXlatorUtils.h`
 - considers the unit provided by the user and returns the result of recalculating the provided number based on the defined base unit
- `HABS_System.cpp`, `HABS_System.h`
 - translates the system level hardware description from the xml file to the memory
- `HABS_ProcChip.cpp`, `HABS_ProcChip.h`
 - translates the processor chip level hardware description from the xml file to the memory
- `HABS_MemChip.cpp`, `HABS_MemChip.h`
 - translates the memory chip level hardware description from the xml file to the memory

- HABS_CConnect.cpp, HABS_CConnect.h
 - translates the chip connectivity level hardware description from the xml file to the memory
- HABS_CoreCluster.cpp, HABS_CoreCluster.h
 - translates the core cluster level hardware description from the xml file to the memory
- HABS_Ip.cpp, HABS_Ip.h
 - translates the hardware IP level hardware description from the xml file to the memory
- HABS_OCMem.cpp, HABS_OCMem.h
 - translates the on-chip memory hardware description from the xml file to the memory
- HABS_Core.cpp, HABS_Core.h
 - translates the processor core hardware description from the xml file to the memory
- HABS_InCuMem.cpp, HABS_InCuMem.h
 - translates the in-cluster memory hardware description from the xml file to the memory
- HABS_OCCConnect.cpp, HABS_OCCConnect.h
 - translates the on-chip connectivity hardware description from the xml file to the memory
- HABS_InCoMem.cpp, HABS_InCoMem.h
 - translates the in-core memory hardware description from the xml file to the memory

4 HABS XML parser

4.1 HABSXlator

4.2 HABSXlatorUtils

5 HABS translator

HABS translator tool translates the parsed information into the memory. It reads the machine description written on the xml file and translate it into the memory. It can also count the number of specific hardware componenets, and calculate some additional information that will provide more detailed information than the xml file. For example, once the user types in the detailed machine description, HABS translator tool can estimate the performance or power based on the description.

Each different hardware component is implemented as a class, which I will continue describing below. The tool set is written in C++, and compiled with g++ 4.2.1.

5.1 System

5.1.1 functions

Synopsis: #include "HABS_System.h"

- int isOpen()
- void close()
- char *get_Name()
- char *get_Comment()
- unsigned int get_NProcChips()
- unsigned int get_NMemChips()
- unsigned int get_NProcCores()
- void est_MemCap()
- long double get_MemCap()
- unsigned int get_MaxSystemPerf(char *ClockFreq)
- long double get_SystemPower(int index)

5.2 Processor Chip

5.2.1 functions

Synopsis: #include "HABS_ProcChip.h"

- char *get_Name()
- char *get_Comment()
- unsigned int get_NClusters()
- char *get_ClusterName(unsigned int ClusterNum)
- char *get_CoreType(unsigned int ClusterNum)
- unsigned int get_NHWIPs()
- char *get_HWIPName(unsigned int HWIPNum)
- unsigned int get_NOCMem()
- char *get_OCMemName(unsigned int OCMemNum)
- char *get_OCMemType(unsigned int OCMemNum)
- unsigned int get_NDMA()
- char *get_DMASrc(int DMANumber)
- char *get_DMADst(int DMANumber)
- long double get_MaxSpeed()

- unsigned int get_MaxChipPerf(char *ClockFreq)
- long double get_ChipPower(unsigned int index)
- char *get_ChipPowerManagement()
- long double get_LeakageCurrent()
- bool get_PowerDownMode()
- long double get_PowerDownCurrent()
- long double get_Voltage()

5.3 Memory Chip

5.3.1 functions

Synopsis: #include "HABS_MemChip.h"

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- long double get_Capacity()
- unsigned int get_NBanks()
- unsigned int get_NRows()
- unsigned int get_NCColumns()
- unsigned int get_WordWidth()
- unsigned int get_AddrWidth()
- unsigned int get_DataWidth()
- long double get_MinCycle()
- unsigned int get_MinLatency()
- bool get_PowerDownMode()
- long double get_PowerDownCurrent()
- long double get_Voltage()
- long double get_EnergyRead()
- long double get_EnergyWrite()
- long double get_LeakageCurrent()

5.4 Chip Connectivity

5.4.1 functions

Synopsis: #include “HABS_CConnect.h”

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- unsigned int get_NChipConnections()
- unsigned int get_AddrWidth(unsigned int ChipConnectionNum)
- unsigned int get_DataWidth(unsigned int ChipConnectionNum)
- bool get_Shared(unsigned int ChipConnectionNum)
- unsigned int get_DegreeSharing(unsigned int ChipConnectionNum)
- void translation()
- long double get_Speed(unsigned int ChipConnectionNum)
- long double get_LoadCap(unsigned int ChipConnectionNum)
- long double get_Voltage(unsigned int ChipConnectionNum)

5.5 Core Cluster

5.5.1 functions

Synopsis: #include “HABS_CoreCluster.h”

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- char *get_MPSSupport()
- unsigned int get_NProcCores()
- bool get_OSRunnable()
- char *get_NetworkTopology()
- bool get_CacheCoherent()
- unsigned int get_NInCuMem()
- unsigned int get_NDMA()
- char *get_DMASrc(int DMANumber)
- char *get_DMADst(int DMANumber)
- long double get_MaxInBandwidth()

- long double get_MaxSpeed()
- unsigned int get_ClusterPerf(char *ClockFreq)
- char *get_PowerManagement()
- double get_Voltage()
- bool get_PowerDownMode()

5.6 Hardware IP

5.6.1 functions

Synopsis: #include “HABS_Ip.h”

- char *get_Name()
- char *get_Comment()
- char *get_DataInput()
- char *get_DataOutput()
- long double get_DataInputBW()
- long double get_DataOutputBW()
- unsigned int get_DataProcessingLat()
- long double get_MinClockFreq()
- long double get_MinVoltage()
- long double get_MaxClockFreq()
- long double get_MaxVoltage()
- long double get_Capacitance()
- long double get_LeakageCurrent()
- long double get_Voltage(char *ClockFreq)
- long double get_Power(char *ClockFreq)
- bool get_PowerDownMode()
- long double get_PowerDownCurrent()

5.7 On-chip Memory

5.7.1 functions

Synopsis: #include “HABS_OCMem.h”

- char *get_Name()
- char *get_Comment()

- char *get_Type()
- long double get_Capacity()
- unsigned int get_NBanks()
- unsigned int get_NRows()
- unsigned int get_NColumns()
- unsigned int get_WordWidth()
- unsigned int get_NSets()
- unsigned int get_NWays()
- long double get_BlockSize()
- char *get_ReplacementPolicy()
- unsigned int get_AccessLatency()
- long double get_EnergyRead()
- long double get_EnergyWrite()
- long double get_LeakageCurrent()

5.8 Processor Core

5.8.1 functions

Synopsis: #include "HABS_Core.h"

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- unsigned int get_NThreads()
- unsigned int get_NIssueWidth()
- unsigned int get_NFUTypes()
- char *get_FUTypeName(unsigned int FUTypeNum)
- unsigned int get_NInCoMem()
- bool get_ICache()
- bool get_DCache()
- unsigned int get_NominalIPC()
- unsigned int get_NMemPort()
- unsigned int get_MIPS(char *ClockFreq)
- unsigned int pass_MIPS()

- long double get_MinClockFreq()
- long double get_MinVoltage()
- long double get_MaxClockFreq()
- long double get_MaxVoltage()
- long double get_Capacitance()
- long double get_LeakageCurrent()
- long double get_Voltage(char *ClockFreq)
- long double get_Power(char *ClockFreq)
- bool get_PowerDownMode()
- long double get_PowerDownCurrent()

5.9 In-cluster Memory

5.9.1 functions

Synopsis: #include "HABS_InCuMem.h"

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- long double get_Capacity()
- unsigned int get_NBanks()
- unsigned int get_NRows()
- unsigned int get_NCColumns()
- unsigned int get_WordWidth()
- unsigned int get_NSets()
- unsigned int get_NWays()
- long double get_BlockSize()
- char *get_ReplacementPolicy()
- unsigned int get_AccessLatency()
- long double get_EnergyRead()
- long double get_EnergyWrite()
- long double get_LeakageCurrent()

5.10 On-chip Connectivity

5.10.1 functions

Synopsis: #include "HABS_ProcChip.h"

- char *get_Name()
- char *get_Comment()
- char *get_Topology()
- unsigned int get_AddrWidth()
- unsigned int get_DataWidth()
- bool get_CacheCoherent()
- unsigned int get_NConnections()
- unsigned int get_OneAddrWidth(unsigned int ConnectionNum)
- unsigned int get_OneDataWidth(unsigned int ConnectionNum)
- void trans_OneSpeed()
- long double get_OneSpeed(unsigned int ConnectionNum)
- double get_Speed()
- unsigned int get_SrcOverhead()
- unsigned int get_DstOverhead()
- unsigned int get_SwitchDelay()
- long double get_MinClockFreq()
- long double get_MinVoltage()
- long double get_MaxClockFreq()
- long double get_MaxVoltage()
- long double get_Capacitance()
- long double get_LeakageCurrent()
- long double get_Voltage(char *ClockFreq)
- long double get_Power(char *ClockFreq)
- bool get_PowerDownMode()
- long double get_PowerDownCurrent()

5.11 In-core Memory

5.11.1 functions

Synopsis: #include “HABS_InCoMem.h”

- char *get_Name()
- char *get_Comment()
- char *get_Type()
- long double get_Capacity()
- unsigned int get_NBanks()
- unsigned int get_NRows()
- unsigned int get_NColumns()
- unsigned int get_WordWidth()
- unsigned int get_NSets()
- unsigned int get_NWays()
- long double get_BlockSize()
- char *get_ReplacementPolicy()
- unsigned int get_AccessLatency()
- long double get_EnergyRead()
- long double get_EnergyWrite()
- long double get_LeakageCurrent()