



IST ACOTES Project  
Deliverable D2.3

# Machine Description file for the ASM

IST-034869

Public

<b>Project Number</b>	:	IST-034869
<b>Project Title</b>	:	Advanced Compiler Technologies for Embedded Streaming
<b>Deliverable Type</b>	:	Demonstrator

<b>Deliverable Number</b>	:	D2.3
<b>Title of Deliverable</b>	:	Machine Description file for the ASM
<b>Nature of Deliverable</b>	:	Demonstrator
<b>Internal Document Number/version</b>	:	1
<b>Contractual Delivery Date</b>	:	30 May2008
<b>Actual Delivery Date</b>	:	8 September 2008
<b>WP(s)</b>	:	WP 2 Streaming Programming / Abstract StreamingMachine
<b>Author(s)/Affiliation</b>	:	Paul Carpenter/UPC

## Abstract

Machine description file for the Cell Broadband Engine, used as input by the Abstract Streaming Machine

## Keyword list

ASM Abstract Streaming Machine  
Cell Cell Broadband Engine

# Table of Contents

<b>Table of Contents.....</b>	<b>3</b>
<b>1 Introduction.....</b>	<b>4</b>
<b>2 The Machine Description file for the Cell Broadband Engine .....</b>	<b>5</b>

# 1 Introduction

The Acotes project is defining a Streaming Programming Model (SPM) and an Abstract Streaming Machine (ASM) making possible to develop parallel streaming applications, and analyze their performance for several target architectures. The SPM will allow to express the specific parallel processing features of streaming data applications, and the ASM is to provide a target architecture description to be used by the compiler.

The Abstract Streaming Machine requires a high level description of the architecture in order to be able to map the application efficiently on the platform.

## 2 The Machine Description file for the Cell Broadband Engine

The text below is the machine description of the Cell Broadband Engine for the Abstract Streaming Machine. This description uses Python-like syntax, and the parameters are as defined in Deliverable D2.2, "Report on Streaming Programming Model and Abstract Streaming Machine Description Final Version".

```
# Cell Machine Description file
# $Id: cell_asm.py 1462 2008-08-27 15:47:42Z paul $

from System import *

# Measured costs of acolib communications primitives:
# (currently only implemented for SPE)
# pushAcqCost      = 448      # 140ns * 3.2GHz
# pushSendFixedCost = 1104    # 345ns * 3.2GHz
# pushSendUnit     = 16384   # Push unit size in bytes (max. DMA size)
# pushSendUnitCost = 352     # 110ns * 3.2GHz
# popAcqFixedCost  = 317     # 99ns * 3.2GHz
# popDiscCost      = 189     # 59ns * 3.2GHz
# pushSendLatency  = 552     # 345ns * 1.6GHz latency on EIB

# Define platform
def setup_platform():

    # Define processors and memories
    resources = [ # Processors:          acolib parameters (see above)          GHz local
memories
        Processor ( 1, 'PPE', 0, 0, 0, 0, 0, 0, 0, 0, 3.2, [ ]),
        Processor ( 2, 'SPE0', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS0',
0x0) ]),
        Processor ( 3, 'SPE1', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS1',
0x0) ]),
        Processor ( 4, 'SPE2', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS2',
0x0) ]),
        Processor ( 5, 'SPE3', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS3',
0x0) ]),
        Processor ( 6, 'SPE4', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS4',
0x0) ]),
        Processor ( 7, 'SPE5', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS5',
0x0) ]),
        Processor ( 8, 'SPE6', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS6',
0x0) ]),
        Processor ( 9, 'SPE7', 448, 1104, 16384, 352, 317, 0, 0, 189, 3.2, [ ('LS7',
0x0) ]),

        # Memories          size          GHz latency bytes/cycle
        Memory ( 10, 'RAM', 1073741824, 0.4, 4, 8), # 1GB Main memory (set
characteristics,
                                                # if required: example
400MHz)

        Memory ( 11, 'LS0', 262144, 3.2, 2, 128), # Local stores...
        Memory ( 12, 'LS1', 262144, 3.2, 2, 128),
        Memory ( 13, 'LS2', 262144, 3.2, 2, 128),
        Memory ( 14, 'LS3', 262144, 3.2, 2, 128),
        Memory ( 15, 'LS4', 262144, 3.2, 2, 128),
        Memory ( 16, 'LS5', 262144, 3.2, 2, 128),
        Memory ( 17, 'LS6', 262144, 3.2, 2, 128),
        Memory ( 18, 'LS7', 262144, 3.2, 2, 128) ]

    # Define interconnects
    links = [ # EIB
        Link ( 1,
            'EIB',
            ['PPE', 'SPE0', 'SPE1', 'SPE2', 'SPE3', 'SPE4', 'SPE5', 'SPE6', 'SPE7',
'RAM'],
            0, # startCost
            16, # bandwidth: 16 bytes per cycle at 1.6GHz
```

```
    0,          # finCost:
    3,          # numChannels: EIB modelled as a 3-channel bus
    552,        # startLatency
    1.6,        # clockRateGHz
    True ),    # multiplexable

    # Connect each SPE to its local store
    Link ( 2, 'SPE0-LS0', ['SPE0','LS0'], 0,0,0),
    Link ( 3, 'SPE1-LS1', ['SPE1','LS1'], 0,0,0),
    Link ( 4, 'SPE2-LS2', ['SPE2','LS2'], 0,0,0),
    Link ( 5, 'SPE3-LS3', ['SPE3','LS3'], 0,0,0),
    Link ( 6, 'SPE4-LS4', ['SPE4','LS4'], 0,0,0),
    Link ( 7, 'SPE5-LS5', ['SPE5','LS5'], 0,0,0),
    Link ( 8, 'SPE6-LS6', ['SPE6','LS6'], 0,0,0),
    Link ( 9, 'SPE7-LS7', ['SPE7','LS7'], 0,0,0)
]

return Platform ( resources, links )
```