Institute for System Programming of the Russian Academy of Sciences

nML Reference Manual (UNDER DEVELOPMENT)

Moscow 2016

Contents

Chapter 1

Chapter

1.1 Introduction

1.2 Constants

1.3 Data Types

A data type specifies the format of values stored in registers or memory. nML supports the following data types:

- int(N): N-bit signed integer data type. Negative numbers are stored in two's complement form. The range of possible values is [-2n-1 ... 2n-1 - 1].
- card(N): N-bit unsigned integer data type. The range of possible values is [0 ... 2n - 1].
- float(N, M): IEEE 754 floating point number, where fraction size is N and exponent size is M. The resulting type size is N + M + 1 bits, where 1 is an implicitly added bit for store the sign. Supported floating-point formats include:
  - 32-bit single-precision. Defined as float(23, 8).
  - 64-bit double-precision. Defined as float(52, 11).
  - 80-bit double-extended-precision. Defined as float(64, 15).
  - 128-bit quadruple-precision. Defined as float(112, 15).

nML allows declaring aliases for data types. Here is a simple type declaration:

```
type DWORD = card(32)
```

In this example, type is a reserved word, DWORD is the declared alias type name, the card(32) is the actual data type.

```
type HWORD = card(16)
type DWORD = card(64)

type INT   = int(32)
type LONG  = int(64)

reg GPR [32, DWORD]

mode R (i : card(5)) = GPR[i]
  syntax = format("r%d", i)

op lui (rt: R, immediate: HWORD)
  syntax = format("lui␣%s,␣0x%x", rt.syntax, immediate)
  action = {
    rt = coerce(LONG, (coerse(INT, immedidate) << 16));
  }
```