**PE配置规则表**

数据流终点即为：SE/break/（循环依赖的PE？会由数据流连向外层）

|  |  |  |  |
| --- | --- | --- | --- |
| 字段序号 | 配置字含义 | 被解析成数字个数及序号 | 各个数字的含义 |
| 1 | [type+index] | 2（0,1） | 第一个数字：  当前部件类号   1. PE 2. LC 3. LSE   第二个数字：  当前部件标号 |
| 2 | [opcode] | 1（2） | 0—不执行任何操作(dor)  1—有符号加法(add)  2—有符号乘法(mul)  3—有符号减法(sub)  4—除法(div)  5—求模（mod）  6—小于(comb)  7—等于(come)  8—MUX (mux)  9—乘累加(mac)  10—+1(add1)  12—左移(leshift)  13—右移(rishift)  14—按位与(and)  15—按位或(or)  16—(cos)  17—(sin)  18—(hcomb)  19—(hadd)  20—(hmux)  21—(xor)  22—(smux)  注1 |
| 3 | [in0\_from+in0\_port+in0\_flag] | 3（3,4） | 第一个数字：  来源部件的标号  第二个数字  来源部件的类别号  0—PE  1—LC  2—LSE |
| 4 | [in1\_from+in1\_port+in1\_flag] | 3（5,6） | 第一个数字：  来源部件的标号  第二个数字  来源部件的类别号  0—PE  1—LC  2—LSE |
| 5 | [in2\_from+in2\_port+in2\_flag] | 3（7,8） | 第一个数字：  来源部件的标号  第二个数字  来源部件的类别号  0—PE  1—LC  2—LSE |
| 6 | [lr\_from] | 1（9） | 0—悬空flr  1—in1  2—alu |
| 7 | [alu\_in0\_from] | 1（10） | 0—fin  1—ib0  2—lr  3—in1 |
| 8 | [alu\_in1\_from] | 1（11） | 0—fin  1—ib1  2—lr  3—in2 |
| 9 | [alu\_in2\_from] | 1（11） | 0—fin  1—ib2  2—reg\_v  3—in3 |
| 9 | [ob0\_from] | 1（12） | 0—悬空fob1  1—alu  2—lr  3—ib1 |
| 10 | [output0\_from] | 1(13) | 0—ob  1—alu |
| 11 | [control\_mode]  **如何解释bool输入** | 1(14) | 1. Dataflow   **如mux，bool输入是dataflow一部分**   1. Loop activae 2. Loop reset 3. Break\_pre控制 4. Break\_post控制 5. Continue控制 6. Condition控制 |
| 13 | [dfg\_end]  **全局数据流终点** | 1(16) | 1. nop 2. end |

对于普通循环交互和循环依赖，共同点是lc信号控制reg向外输出，区别是前者输出到ob，而后者输出到mux的aluin1

**LC配置规则表**

**//13输入4输出**

|  |  |  |  |
| --- | --- | --- | --- |
| 配置字段序号 | 配置字 | 被解析成数字个数 | 各个数字的含义 |
| 1 | [category+index] | 2(0,1) | 第一个数字：  当前部件类号  0—PE  1—LC  2—LSE  第二个数字：  当前部件标号 |
| 2 | [Outermost] | 1(2) | 0—nop  1—outermost |
| 3 | [Iterator\_From] | 2(3,4) | 第1个数字：  单元标号index  第2个数字：  单元类别  0—PE  1—LC  2—LSE  3—begin |
| 4 | [Initial\_From] | 2(5,6) | 第一个数字：  来源部件的标号  第二个数字  来源部件的类别号0—PE  1—LC  2—LSE |
| 5 | [LoopBound\_From] | 2(7,8) | 第一个数字：  来源部件的标号  第二个数字  来源部件的类别号  0—PE  1—LC  2—LSE |
| 6 | [Step\_From] | 2(9,10) | 第一个数字：  来源部件的标号  第三个数字  来源部件的类别号  0—PE  1—LC  2—LSE |
| 7 | [End\_From]  **此配置同时代表了dfg\_end以及break; dfg\_end使用11进行与操作，break使用10进行或操作** | 2(11, 12) | 第1个数字：  单元标号index  第2个数字：  单元类别  0—PE  1—LC  2—LSE |

其中字段序号7的配置有多个，即end来源为多个，具体个数未定。

**LSE配置规则表**

|  |  |  |  |
| --- | --- | --- | --- |
| 配置字段序号 | 配置字 | 被解析成数字个数 | 各个数字的含义 |
| 1 | [category+le\_index] | 2(0,1) | 第一个数字：  当前部件类号  0—PE  1—LC  2—LSE  第二个数字：  当前部件标号 |
| 2 | [in0\_from+in0\_port] | 2(2,3) | 第1个数字：  单元标号index  第2个数字：  单元类别  0—PE  1—LC  2—LSE |
| 3 | [in1\_from+in1\_port] | 2(4,5) | 第1个数字：  单元标号index  第2个数字：  单元类别  0—PE  1—LC  2—LSE |
| 4 | [Load/Store] | 1(6) | 0—load  1—storeA  2—storeD |
| 5 | [tagBind] | 1(7) | 0—null  1—tag |
| 6 | [daeMode] | 1(8) | 0—cache  1—dae |
| 7 | [bufferMode] | 1(9) | 0—fifo  1—ram |
| 8 | [fifoStep] | 4(10,11,12,13) | Ptr\_step |
| 9 | [bufferSize] | 2(14,15) | small—s  middle—m  large—l |
| 10 | [match] | 3(16,17,18) | 1. !match 2. match |