# PerformanceCounter 设计方案一

## 统计方式设计

统计事件按照性能分析场景进行分组，每个group下最多可统计16个事件。比如pipeline\_balance\_static的分组下是对流水各阶段指令流动是否均衡的统计。当有新的性能分析需求就增加新的group，即对group进行拓展。

Group拓展时counter计数器的数目保持不变（始终是16个），需要做出调整的硬件资源有：

1. event\_ctl SPR(special purpose register)寄存器的group\_sel字段进行拓宽。

2.每个event的信号来源增加1bit

|  |  |  |  |
| --- | --- | --- | --- |
| EVENT\_GROUP | Group\_num | EVENT | Counter\_Addr |
| Pipeline\_balance\_static | 0 | DISPATCH\_WIDTH | 0 |
| PREDICTION | 1 |
| … | 2 |
|  | … |
|  | 15 |
| Fetch\_performance\_static | 1 |  | 0 |
|  | 1 |
|  | 2 |
|  | … |
|  | 15 |

## 计数硬件设计



如上图所示，总共有16个counter， 地址分别是0到15， 每个counter有一个en作为计数使能，同时计数事件来源通过group\_sel来进行选择，选择要进行分析的性能组。En和count\_en来自event\_ctl SPR 寄存器。

## 相关SPR寄存器

增加三个SPR寄存器用于性能统计，分别是event\_ctl， event\_access, event\_data.

**EVENT\_CTL**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 |  |  |  |  |  |  | 24 | 23 |  |  |  |  |  |  | 16 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
|  | | | | | | | | Group\_sel | | | | | | | | Count\_en | | | | | | | | | | | | | | | |

Count\_en：为1时对应的counter寄存器每遇到一次有效的event事件，counter加1

Group\_sel：选择每个counter计数事件的来源

**EVENT\_ACCESS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 5 | 4 |  |  | 1 | 0 |
| clear |  | | | | | | | | | | | | | | | | | | | | | | | | | | addr | | | | we |

We：write\_en, clean为0， 当对该SPR写入时we=1，将event\_data寄存器中的值写入addr所描述counter寄存器中；当clean为0， we=0时，将addr所描述的counter寄存器的值写入event\_data寄存器中。

Addr：用于描述所访问counter寄存器地址。

Clean：当event\_access的写入值的we=1， clean=1时，将所有counter寄存器的值清零。

**EVENT\_DATA**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Data：用于保存couner寄存器的读出或者写入值。

## 性能统计流程

1.启动整个group从0开始计数

写event\_ctl, 所有en为0

写event\_access寄存器，clean=1, we=1

写event\_ctl寄存器，count\_en=0xffff, group\_sel=group\_num

2.启动group中某个event计数

写event\_data寄存器，设置要计数的初始值

写event\_access寄存器，we=1, clean=0, addr=event\_num

写event\_ctl寄存器，count\_en[event\_num]=1, group\_sel=group\_num

3.读出counter寄存器的值

写event\_ctl寄存器，count\_en=0，停止计数

写event\_access寄存器，we=0， addr=event\_num, counter[event\_num]值被读到event\_data寄存器中

RSETSPR指令将event\_data的值读到GPR

# PerformanceCounter设计方案二

## 统计方式设计

统计事件按照设计功能块进行分组，每个分组下可统计的事件数目可以拓展。Group数目也可以进行拓展。比如icache\_performnce\_static是所有从icache相关模块拉出来的统计信号，当前设置的每个group下的event数目为16。

对group下的event进行拓展时，counter数目保持不变，需要做出调整的硬件资源有：

1. event\_access SPR寄存器的event\_num字段进行拓宽

2. counter寄存器的信号来源数目增加

Group下的event数目保持不变，对group进行拓展时，需要做出调整的硬件资源有：

1.增加一组counter寄存器

2.event\_ctl SPR寄存器增加一组gp\_count\_en信号

3.event\_data SPR寄存器的gp\_num字段进行拓宽

|  |  |  |  |
| --- | --- | --- | --- |
| EVENT\_GROUP | Group\_num | EVENT | Count\_addr |
| ICACHE\_PERFORMANCE\_STATIC | 0 | ITLB\_ACCESS | 0 |
| ITLB\_MISS | 1 |
| ICACHE\_MISS | 2 |
| … | … |
| … | 15 |
| IFETCH\_PERFORMANCE\_STATIC | 1 | ROB\_FULL | 0 |
| DISPATCH0 | 1 |
| DISPATCH1 | 2 |
|  | … |
|  | 15 |
|  |  |  |  |

## 计数硬件设计



如上图所示，每个group下有4个counter，每个counter都有计数使能（图中未画出），计数使能来自event\_ctl寄存器对应的grp\_cnt\_en，每个counter都连接group下的所有事件。由event\_num来选择计数信号来源。

相关SPR寄存器

## 相关SPR寄存器

**EVENT\_CTL**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 8 | 7 |  |  | 4 | 3 |  |  | 0 |
|  | | | | | | | | | | | | | | | | | | | | | | | | gp1\_cnt\_en | | | | gp0\_cnt\_en | | | |

Gp0\_cnt\_en：group0的四个counter的计数使能

Gp1\_cnt\_en：group1的四个counter的计数使能

**EVENT\_ACCESS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 12 | 11 |  |  | 8 | 7 |  |  | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | event\_num | | | | grp\_num | | | | cnt\_num | | | |

Cnt\_num：表示读或者写的counter寄存器编号

Grp\_num：表示读或者写的counter寄存器所在group

Event\_num：当we为1时，写event\_access寄存器会将event\_num值写入grp\_num和cnt\_num所描述的counter寄存器中

We：表示对counter寄存器的写操作。当写入event\_access的we的值为0时，会将grp\_num和cnt\_num所描述的counter寄存器的值写入event\_data寄存器中；当写入event\_access的we值为1时， 会将event\_num的值写入grp\_num和cnt\_num所描述的counter寄存器的event\_num字段，并将counter字段清零。

**EVENT\_DATA**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |
| Data | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Data：用于保存couner寄存器的读出或者写入值。

## 性能统计流程

1.启动某个counter的计数

写event\_ctl寄存器，所有en为0

写event\_access寄存器，we=1， event\_num=要统计事件在group下编号

Grp\_num=group编号，cnt\_num=counter寄存器地址。

写event\_ctl寄存器，所有en为1， 开始计数

2.读取某个counter的值

写event\_ctl寄存器，所有en为0， 停止计数

写event\_access寄存器，we=0， grp\_num=group编号，cnt\_num=counter寄存器地址，counter的值会被写入event\_data寄存器中

RSETSPR将event\_data寄存器的值写入GPR