**PartB**

Part要求修改seq-full.hcl添加IADDQ指令，即常数与寄存器相加并将结果存储在寄存器中，IADDQ的五阶段描述如下所示：

* 取指阶段

icode:ifun <- M1[PC]

rA:rB <- M1[PC+1]

valC <- M8[PC+2]

valP <- PC+10

* 译码阶段

valB <- R[rB]

* 执行阶段

valE <- aluA(valC) + aluB(valB) Set CC

* 访存阶段

无

* 写回阶段

R[rB] <- valE(dstE)

* PC更新阶段

PC <- valP

根据五阶段描述，应该在seq-full.hcl文件中做出的修改如下所示：

* 取指阶段

添加IIADDQ作为有效指令：

bool instr\_valid = icode in

{ INOP, IHALT, IRRMOVQ, IIRMOVQ, IRMMOVQ, IMRMOVQ,

IOPQ, IJXX, ICALL, IRET, IPUSHQ, IPOPQ, IIADDQ};

添加IIADDQ需要寄存器：

bool instr\_valid = icode in

{ INOP, IHALT, IRRMOVQ, IIRMOVQ, IRMMOVQ, IMRMOVQ,

IOPQ, IJXX, ICALL, IRET, IPUSHQ, IPOPQ, IIADDQ};

添加IIADDQ需要常数：

bool need\_valC =

icode in { IIRMOVQ, IRMMOVQ, IMRMOVQ, IJXX, ICALL, IIADDQ};

* 译码阶段

添加IIADDQ需要源寄存器rB:

## What register should be used as the B source?

word srcB = [

icode in { IOPQ, IRMMOVQ, IMRMOVQ, IIADDQ} : rB;

icode in { IPUSHQ, IPOPQ, ICALL, IRET } : RRSP;

1 : RNONE; # Don't need register

];

* 执行阶段

在aluA的valC中添加IIADDQ，在aluB中的valB中添加IAADQ

## Select input A to ALU

word aluA = [

icode in { IRRMOVQ, IOPQ } : valA;

icode in { IIRMOVQ, IRMMOVQ, IMRMOVQ, IIADDQ} : valC;

icode in { ICALL, IPUSHQ } : -8;

icode in { IRET, IPOPQ } : 8;

# Other instructions don't need ALU

];

## Select input B to ALU

word aluB = [

icode in { IRMMOVQ, IMRMOVQ, IOPQ, ICALL,

IPUSHQ, IRET, IPOPQ, IIADDQ} : valB;

icode in { IRRMOVQ, IIRMOVQ } : 0;

# Other instructions don't need ALU

];

在需要设置CC中添加IIADDQ：

## Should the condition codes be updated?

bool set\_cc = icode in { IOPQ , IIADDQ};

* 访存阶段

无需修改

* 写回阶段

添加IIADDQ需要目的寄存器rB:

## What register should be used as the E destination?

word dstE = [

icode in { IRRMOVQ } && Cnd : rB;

icode in { IIRMOVQ, IOPQ, IIADDQ} : rB;

icode in { IPUSHQ, IPOPQ, ICALL, IRET } : RRSP;

1 : RNONE; # Don't write any register

];

* PC更新阶段

无需修改