## processor.cfg Se In-Order, diversos parâmetros do arquivo são ignorados // 0 -> 0o0 || 1 -> In-order IN\_ORDER = 1; Para cada etapa do pipeline, uma instrução só é processada se o buffer para a próxima etapa estiver vazio

processor\_t bool IN\_ORDER;

INSTANTIATE\_GET\_SET(bool, IN\_ORDER);

processor\_t::allocate

if (cfg\_processor.exists("IN\_ORDER")) { set\_IN\_ORDER(cfg\_processor["IN\_ORDER"] ? true : false); }

{ set\_IN\_ORDER(false); }

processor\_t::fetch Alteração: if (this->fetchBuffer.size == FETCH\_BUFFER)

if ((this->fetchBuffer.size == FETCH\_BUFFER) ||
 (IN\_ORDER && (this->fetchBuffer.size > 0)))

processor\_t::decode

Após:
 if (this->fetchBuffer.is\_empty())

break; }

Adicionar:
if (IN\_ORDER && (this->decodeBuffer.get\_size() > 0)) this->add\_stall\_full\_DecodeBuffer();
break;
}

processor\_t::rename

Alteração:
if (reorderBuffer.robUsed >= ROB\_SIZE) {

if ((reorderBuffer.robUsed >= ROB\_SIZE) ||
 (IN\_ORDER && (reorderBuffer.robUsed > 0)))

processor\_t::dispatch

Alteração:
 if (total\_dispatched >= DISPATCH\_WIDTH) {
 break;

if ((total\_dispatched >= DISPATCH\_WIDTH) ||
 (IN\_ORDER && (total\_dispatched > 0))) break;

Alteração:
if (uop\_total\_executed == EXECUTE\_WIDTH) break;

if ((uop\_total\_executed == EXECUTE\_WIDTH) ||
 (IN\_ORDER && (uop\_total\_executed > 0)))

processor\_t::commit

Alteração: for(uint32\_t i = 0; i < COMMIT\_WIDTH; i++)

Para:
uint32\_t limit = (IN\_ORDER) ? 1 : COMMIT\_WIDTH;
for(uint32\_t i = 0; i < limit; i++)</pre>

• • •

Aqui vai enviar uma a uma porque elas dependem do registrador que liga as uops de uma instrução :p