

processor_t::decode()

```
Ajustes
...
uint32_t last_generated_uop = uops.size() - 1;
uint32_t current_uop_counter = 0;
// Iterate over ups from instruction
for (uint32_t uop_idx : uops)
{

    ...
    if (instr->num_reads > 0)
    {
        ...
    }
    // Dependency with previous uop from instruction
    else if (current_uop_counter > 0) {
        // Find the first empty register
        for (uint32_t i = 0; i < MAX_REGISTERS; i++)
        {
            if (new_uop.read_regs[i] == POSITION_FAIL)
            {
                // Use it to link
                new_uop.read_regs[i] = UOPS_LINK_REGISTER;
                break;
            }
        }
    }

}

if ((instr->num_writes > 0) || (instr_op == INSTRUCTION_OPERATION_BRANCH)
    || (current_uop_counter < last_generated_uop))
{
    ...
}
...
current_uop_counter++;
} // fim for (uint32_t uop_idx : uops)
```

Descrição do problema:
Apesar das uops terem dependências com uops de load/store/branch da mesma instrução, elas estão independentes de outras uops de operações.

Assim, uma instrução MultiplyAdd iria gerar duas uops independentes (multiplicação e soma).