-Data Memory and Instruction Memory are separated

-Exceptions in data memory

-Mux to define value of adder used on PC whether 16 bit or 32 bit or zero

No Operation -> Normal instruction

Hlt -> adding zero to pc

SetC -> Decode Stage

INC -> Control Unit sends signal to ALU

~~Out -> Execute, 2~~~~nd~~ ~~Half Cycle~~

~~In -> Execute to Write Back~~

~~Updating flags -> Execute~~

~~Push and Pop -> In Memory Stage with changing SP with adder~~

~~JMP Decision -> on ALU so it doesn’t make a problem with flag (Assume not Taken)~~

~~Call -> ALU (PC+1) , Mem ( push stack, sub sp , update pc), add nop until finished~~

~~RET -> update sp inside Mem , add NOP until finished software solution #3 nop~~

~~Int -> creating a new flag register to store the values of the flags in case of int~~