Vo Hieu Nghia

ADDRESSING MODES ASSIGMENT

Exercise1:

STR <Rd>, [<Rn>, offset]

ex: str r2, [r3, #4]

STRH <Rd>, [<Rn>, offset]

ex: strh r2, [r3, #12]

STR.W <Rxf>, [<Rn>, offset]

ex: str.w r12, [r7, #4]

MOV SP, Rm (copy)

ex:mov sp, r7

MOV{S} Rd, <Operand2>

ex:movs r3, #0

LDR <Rd>, [<Rn>, #<imm>]

ex:ldr r2, [r7, #28]

LDRH <Rd>, [<Rn>, #<imm>]

ex:ldrh r3, [r3, #0]

LDRSH.W <Rxf>, [<Rn>, offset]

ex:ldrsh.w r3, [r7, #26]

(*data types: word, half word)

Ex2:

When the bx instruction was deleted, the debug jumped from pop line to the next one. However, if the bx instruction is there, then the debug would jump from pop line to the asm_test() where the function had been called.

The reason is that bx is branch and exchange instruction set, which means that according to the lsb (least significant bit) of the address to branch, the processor will treat the next instruction as ARM or as thumb. In addition, lr usually holds the return address, it means that this is a return from a function, it will treat the code at that address as thumb, otherwise, it will treat it as ARM.