Causes of interrupts:

|  |  |
| --- | --- |
| **Types** | **Causes** |
| Overflow | INC – DEC – ADD – IADD – SUB – PUSH – POP – CALL – RET – RTI – PC might overflow every new cycle |
| Out of bounds | LDD – STD – JZ – JC – JMP – CALL – RET – RTI |

Elaboration:

1. INC:
   1. R[Rsrc1] + 1 overflows 🡪 Execution stage
2. DEC:
   1. R[Rsrc1] – 1 overflows 🡪 Execution stage
3. ADD:
   1. Overflow 🡪 Execution stage
4. IADD:
   1. Overflow 🡪 Execution stage
5. SUB:
   1. Overflow 🡪 Execution stage
6. PUSH:
   1. SP -= 1 overflows 🡪 Memory 1 stage
   2. DataMemory[SP] out of bound 🡪 Memory 1 stage (after seeing SP value, if it is greater than 2^10 – 1, then out of bound)
7. POP:
   1. SP += 1 overflows 🡪 Memory 1 stage
   2. DataMemory[SP] out of bound 🡪 Memory 1 stage
8. LDD:
   1. DataMemory[R[Rsrc1]] out of bound, as Memory has 2^10 addresses, but R[Rsrc1] can go up to 2^16-1 🡪 Decode stage (after seeing R[Rsrc1] value, if it is greater than 2^10 – 1, then out of bound)
9. STD:
   1. DataMemory[R[Rsrc]] out of bound 🡪 Decode stage
10. JZ:
    1. PC = R[Rdst] may be out of bound, if instruction cache size is smaller than 2^16-1 🡪 Execution stage
11. JC:
    1. PC = R[Rdst] out of bound 🡪 Execution stage
12. JPM:
    1. PC = R[Rdst] out of bound 🡪 Decode stage (when decoding, we know it is a jump command, and we also have the value of R[Rdst], if it is greater than the size of PC, then out of bound) (unlike JZ and JC, we do not need to wait till the execution stage to check whether we will actually jump or not, as in the JPM command, we will unconditionally jump)
13. CALL:
    1. DataMemory[SP] out of bound, as memory has 2^10 addresses, but SP can go up to 2^16-1 🡪 Memory 1 stage
    2. PC + 1 overflows 🡪 Fetch stage
    3. SP -=1 overflows 🡪 Memory 1 stage
14. RET:
    1. SP += 1 overflows 🡪 Memory 1 stage
    2. DataMemory[SP] out of bound 🡪 Memory 1 stage
15. RTI:
    1. SP += 1 overflows 🡪 Memory 1 stage
    2. DataMemory[SP] out of bound 🡪 Memory 1 stage
16. After each fetch, PC + 1 might overflow 🡪 Fetch stage

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Stage at which interrupt is issued | Interrupt | IF Flush | ID Flush | Ex Flush | Mem Flush | SavePC  (2 bits) |
| Decode | 1 | 1 |  |  |  | 00 |
| Fetch | 1 |  |  | 01 |
| Execute | 1 | 1 |  | 10 |
| Memory 1 | 1 | 1 | 1 | 11 |