Register Y – Address of IDR

Register X – Total bytes received

cpl ($14,y) – This should just be a debugging output, do nothing about the receiving algorithm

($01, SP) – A buffer

|  |  |  |  |
| --- | --- | --- | --- |
| Assembly instruction | Accumulator (a) | Buffer | C |
|  | ??? |  |  |
| srl ($01, SP) | ??? |  | 0 |
| ld a,(y) |  |  | 0 |
| jreq L\_End\_Rx ;SE0 Detection |  |  | 0 |
| xor a, ($01, SP) |  |  | 0 |
| ld ($01, SP),a |  |  | 0 |
|  |  |  |  |
| srl ($01, SP) |  |  | 0 |
| ld a,(y) |  |  | 0 |
| jreq L\_End\_Rx ;SE0 Detection |  |  | 0 |
| xor a, ($01, SP) |  |  | 0 |
| ld ($01, SP),a |  |  |  |
|  |  |  |  |
| srl ($01, SP) |  |  |  |
| ld a,(y) |  |  |  |
| jreq L\_End\_Rx ;SE0 Detection |  |  |  |
| xor a, ($01, SP) |  |  |  |
| ld ($01, SP),a |  |  |  |

Intuitively, we can conclude that:

Bn – current received bit

Dn – current bus status

Dn-1 – previous bus status

In order to make the algorithm work, PC0 – PC5 must remain floating =>

After 3 receiving cycles:

The algorithm cannot handle bit-stuffing

The property of XOR being used:

A xor 0 = A

Last bit(Bit 7, MSB):

|  |  |  |  |
| --- | --- | --- | --- |
| Assembly instruction | Accumulator (a) | Buffer | C |
|  | We don’t care |  |  |
| srl ($01, SP) |  |  | B0 |
| ld a,(y) |  |  | B0 |
| jreq L\_End\_Rx ;SE0 Detection |  |  | B0 |
| xor a, ($01, SP) |  |  | B0 |
| ld ($01, SP),a |  |  | B0 |
| rlc a |  |  | D7 |
| ld (Buf,x), a |  |  | D7 |
| ;Next byte |  |  |  |
| srl ($01, SP) |  |  | B1 |
| ld a,(y) |  |  | B1 |
| jreq L\_End\_Rx ;SE0 Detection |  |  | B1 |
| xor a, ($01, SP) |  |  | B1 |
| ld ($01, SP),a |  |  | B1 |