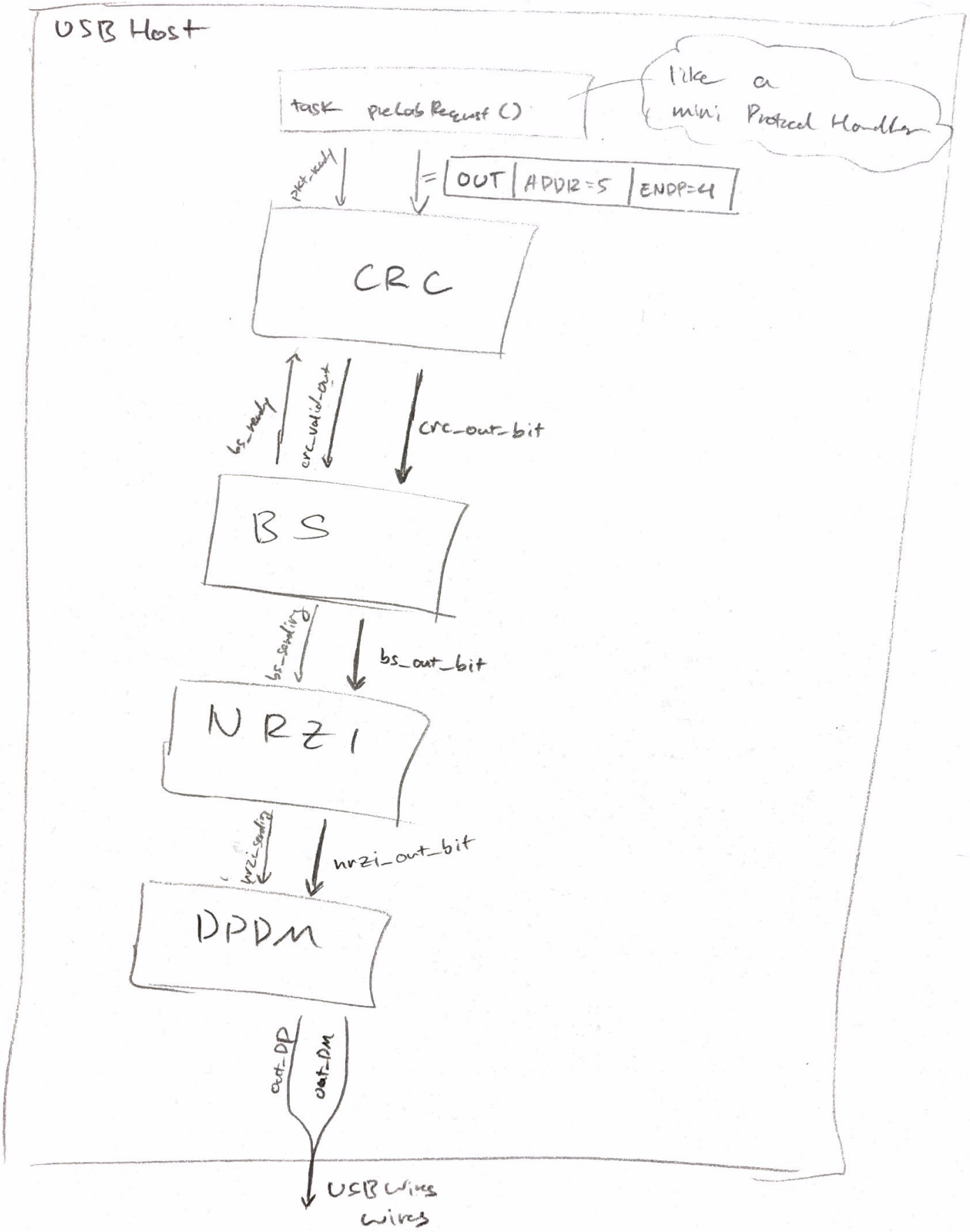
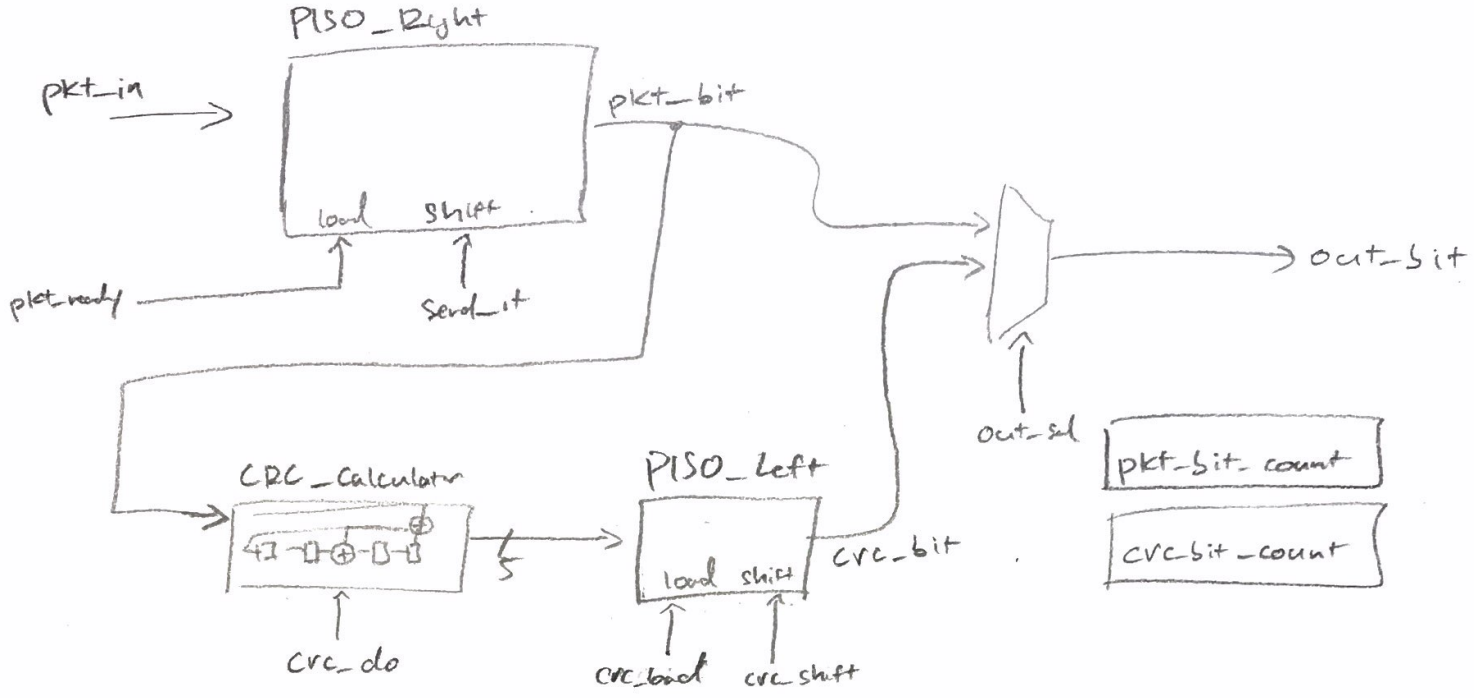
Pic-bus : High-level interconnect of all modules

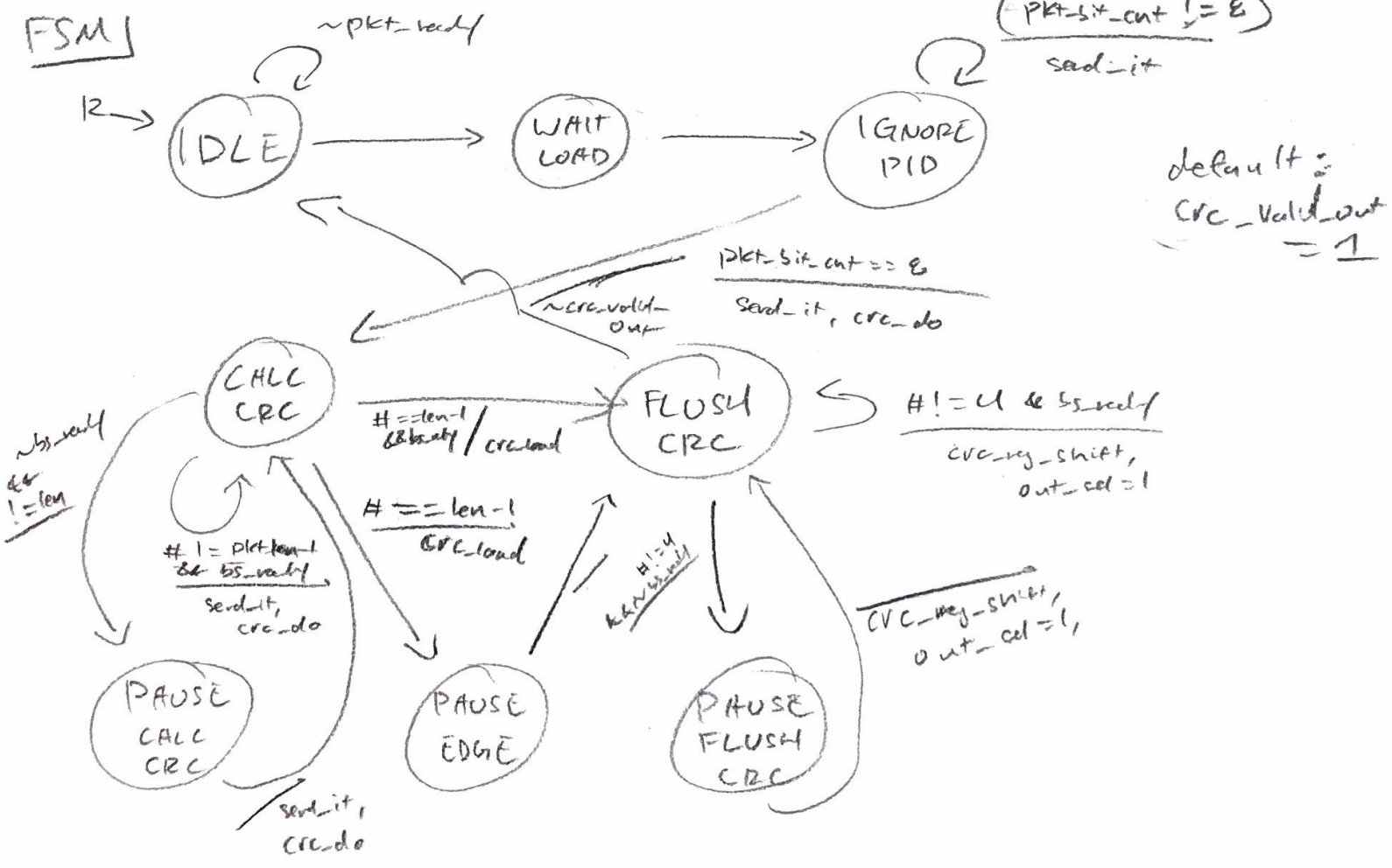


CRC

DP

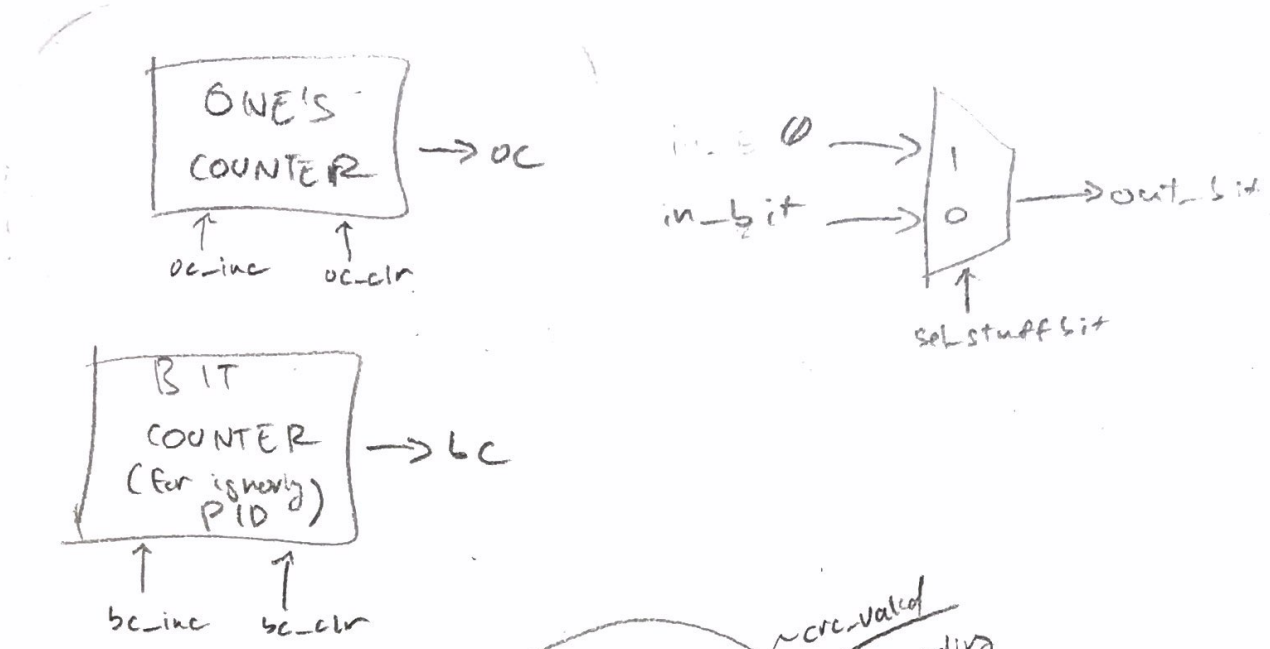

PISO_Right
pkt_in
pkt_bit
load
shift
pkt_ready
send_it
out_bit
out_sel
pkt_bit_count
crc_bit_count
CRC_Calculator
PISO_Left
load
shift
crc_bit
crc_do
crc_load
crc_shift

FSM


~pkt_ready
(pkt_bit_cnt != 8)
send_it
12
IDLE
WAIT LOAD
IGNORE PID
default: crc_valid_out = 1
pkt_bit_cnt == 8
~crc_valid_out
send_it, crc_do
CALC CRC
FLUSH CRC
#!=4 && bs_ready
crc_reg_shift, out_sel=1
~bs_ready && !=len
#==len-1 && bs_ready / crc_load
#==len-1
crc_load
#!=4 && bs_ready
#!=len-1 && bs_ready
send_it, crc_do
crc_reg_shift, out_sel=1,
PAUSE CALC CRC
PAUSE EDGE
PAUSE FLUSH CRC
send_it, crc_do

Bit Stuffer

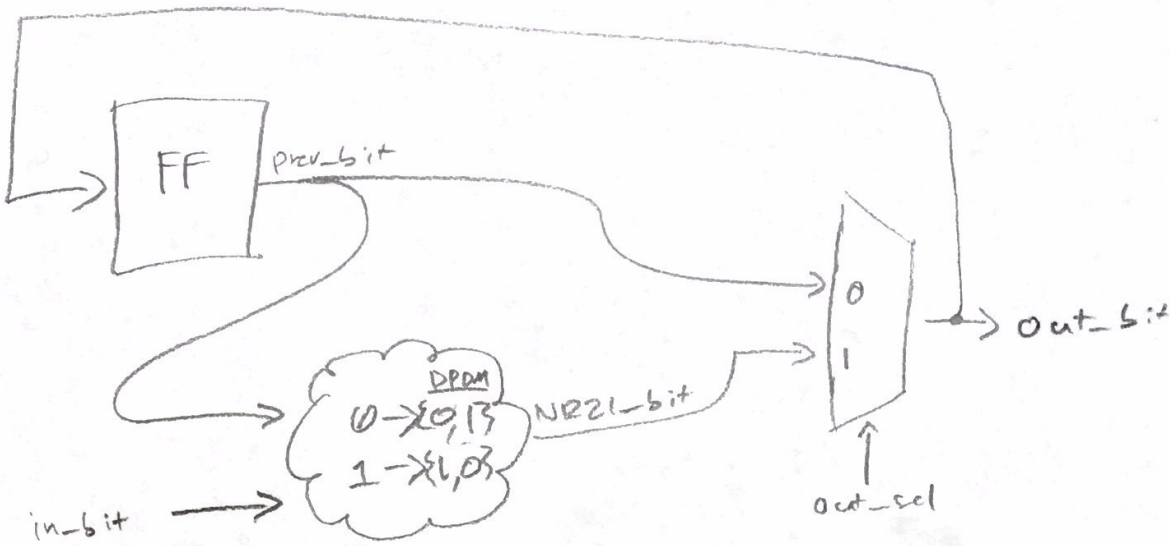
DP



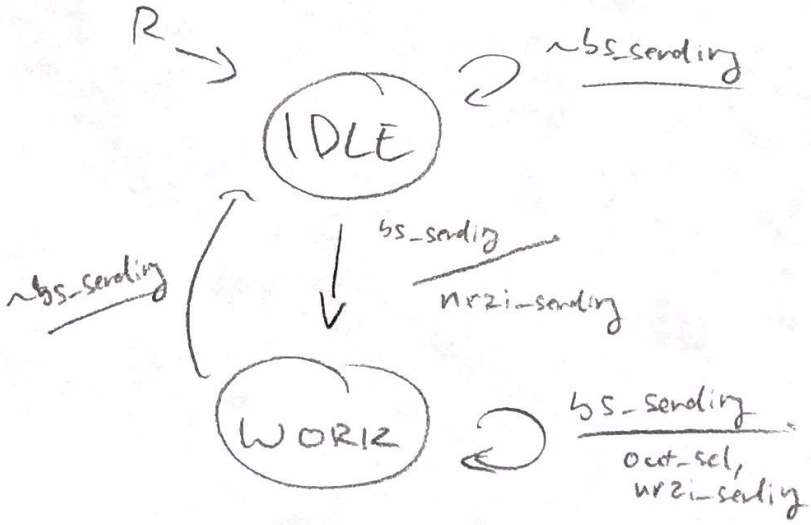
FSM

NRZI

DP

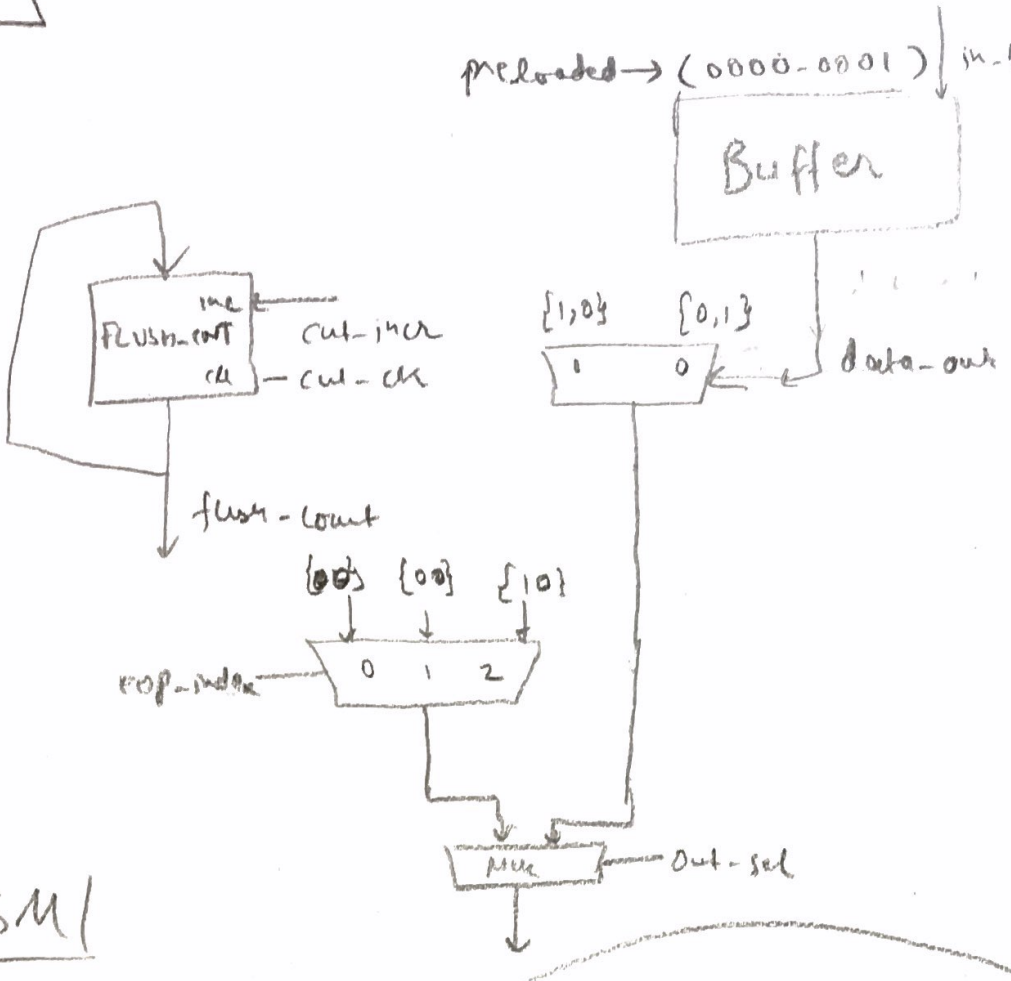

FF
prev_bit
0
1
out_bit
DPDM
0 -> 0,1
1 -> 1,0
NRZI_bit
in_bit
out_sel

FSM

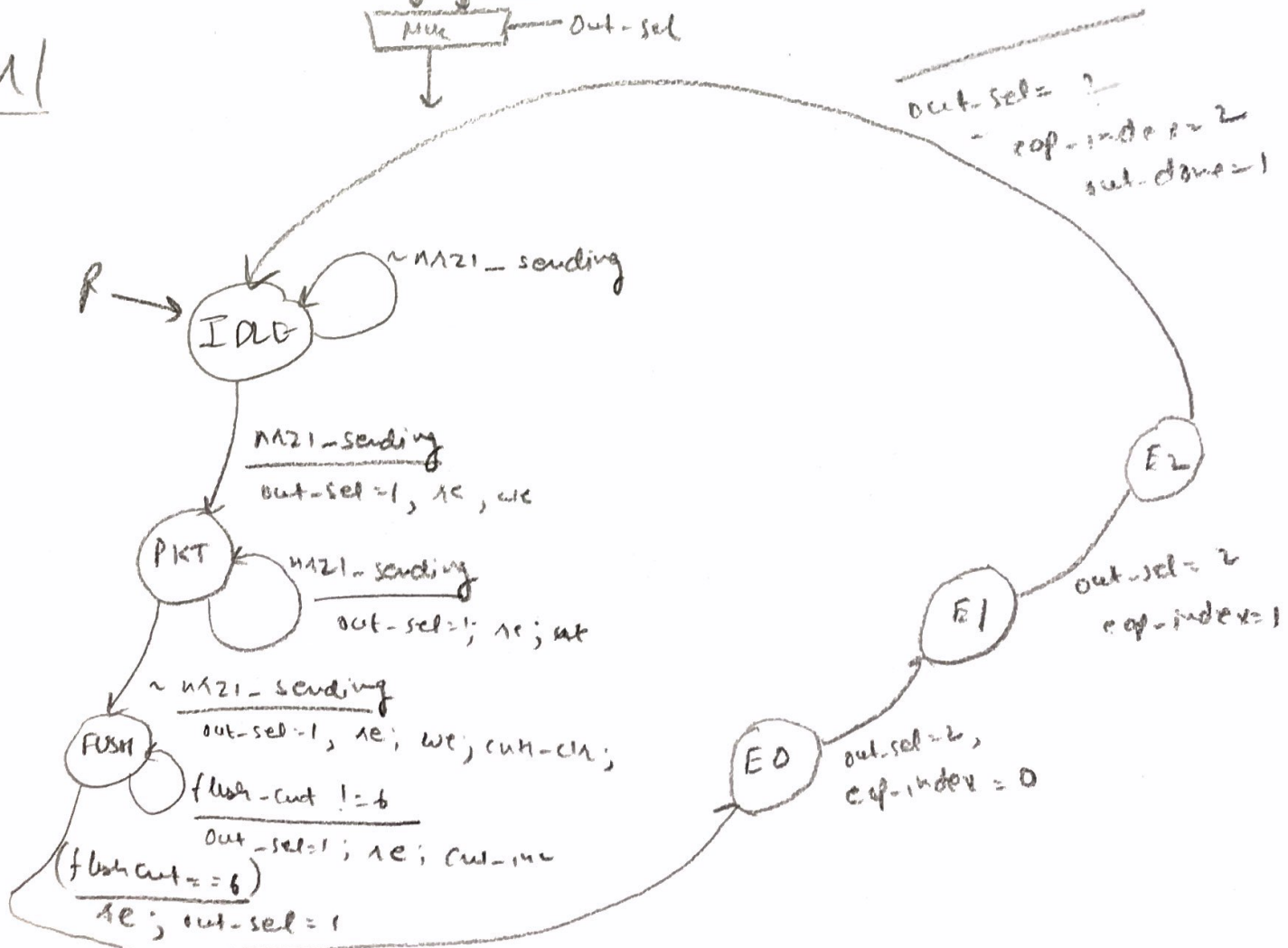

R
IDLE
~bs_sending
bs_sending
nrzi_sending
~bs_sending
WORK
bs_sending
out_sel,
nrzi_sending

DP/DM

DP


preloaded → (0000-0001)
in-bit
Buffer
FLUSH-CNT
inc
clr
cnt-incr
cnt-clr
{1,0}
{0,1}
1
0
data-out
flush-count
{00}
{01}
{10}
0
1
2
rep-index
MUX
out-sel

FSM


out-sel = 2
- rep-index = 2
out-done = 1
R
IDLE
~MAC_sending
MAC_sending
out-sel=1, le, wr
PKT
MAC_sending
out-sel=1; le; wr
~MAC_sending
FLSH
out-sel=1, le; we; cnt-clr;
flush-cnt != 6
out-sel=1; le; cnt-inc
(flush-cnt == 6)
le; out-sel = 1
E0
out-sel=2,
rep-index = 0
E1
out-sel = 2
rep-index=1
E2