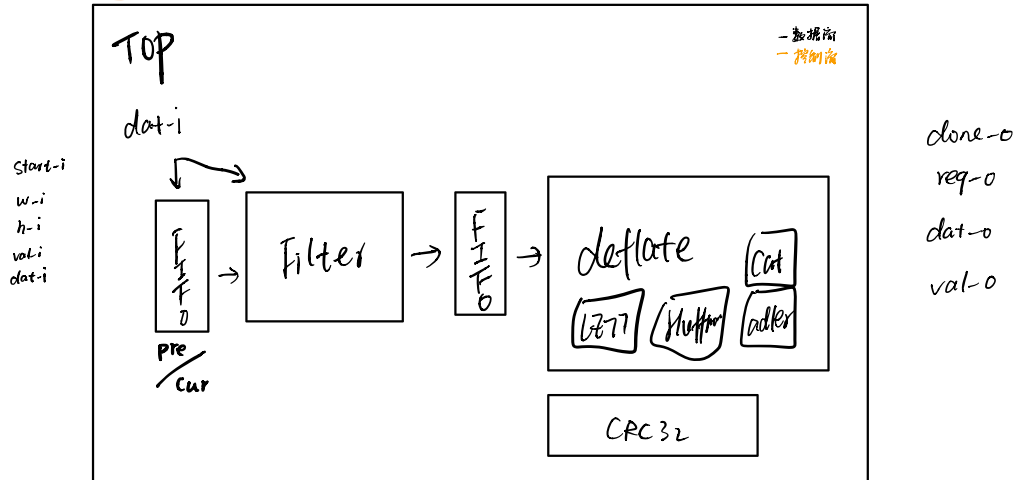
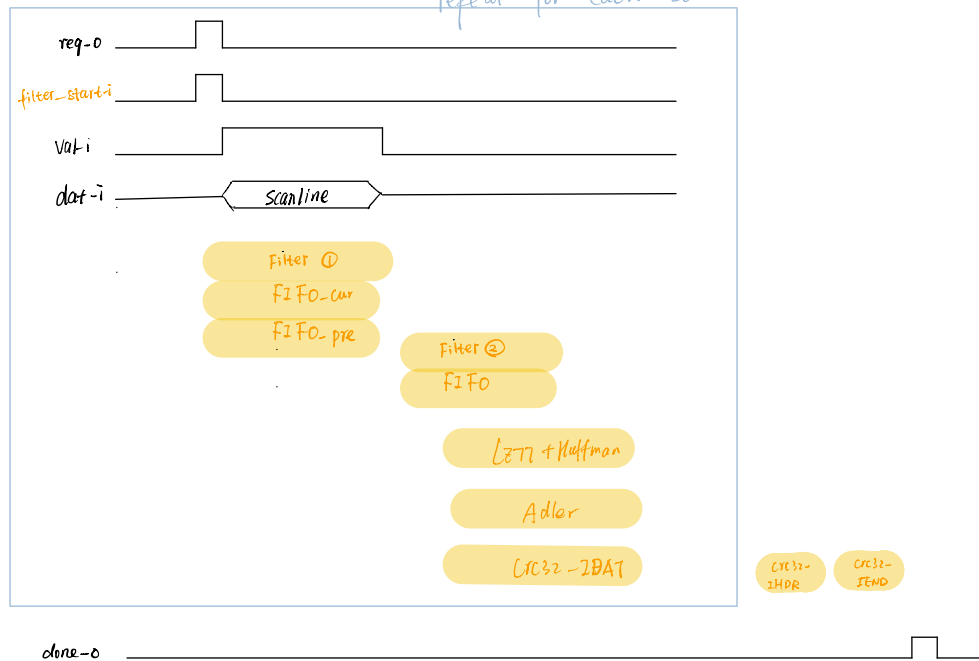


框③

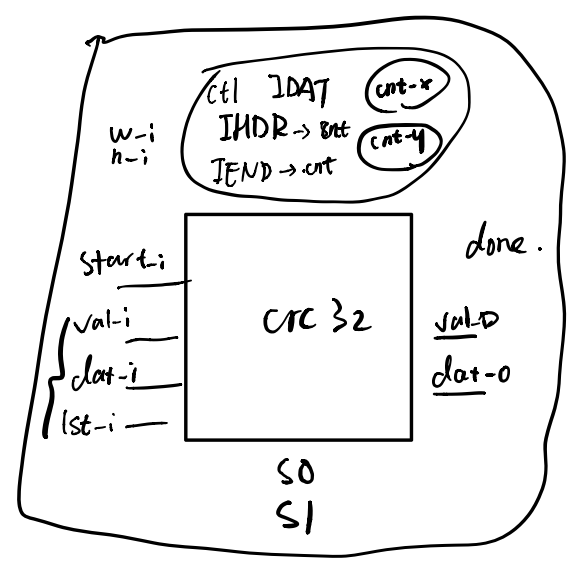
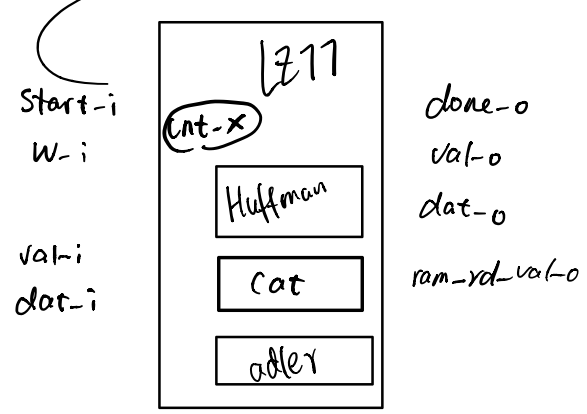
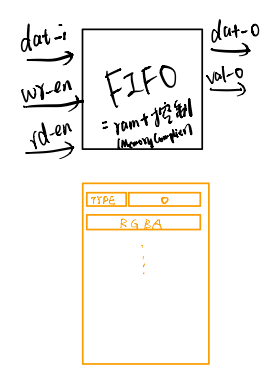
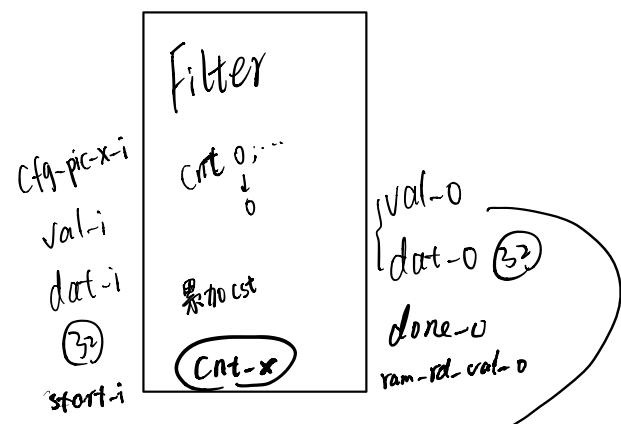
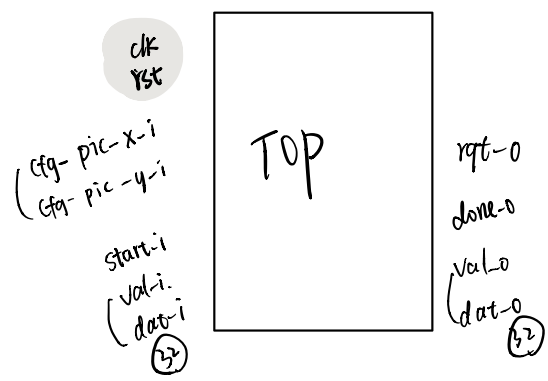


start-i

repeat for each scanline



3 模块



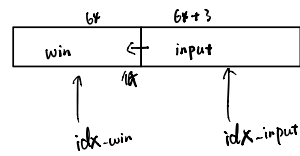
l277

1. UPDATE

cycle: m+1 +1 *fifo Delay*

① sliding window: whole = { win, input }
 $\text{win} \leq \underset{\text{max}}{\text{win}} [- \text{len} : 64]$

cycle: 1



② input:

stage 0: input ← { input, data-i }
 stage 1: input ← input << left-shift
 ↓
 $64 - (x - \text{len} + 4m)$

cycle: $\left\{ \begin{array}{l} 16 \text{ (64/4)} \text{ cnt_pxl} = 0 \\ m, x - \text{len} + 4m \geq 64 \Rightarrow m = \frac{64 - x + \text{len} + 3}{4} \end{array} \right.$
 cycle: 1

2. SEARCH

#define PARALLEL 1/2/4

flg { generate PARALLEL
 $\text{flg} [64 * \text{idx-parallel} + 0] = \text{input} [0] == \text{whole} [0 + \text{idx-parallel}]$
 \vdots
 63

flg- { generate PARALLEL
 $\text{idx} == 0 \rightarrow m(\text{idx})$
 $\text{flg-} [\text{idx} \times 64 + 64] = \text{flg} [\text{idx} \times 64 + 64]$
 $\text{idx} == 1$
 $\text{flg-} [m(\text{idx})] = \text{flg} [m(\text{idx})]$
 $\quad \& \text{flg} [m(\text{idx}-1)]$
 $\text{idx} == \dots$
 $\text{flg-} [m(\text{idx})] = \text{flg-} [m(\text{idx}-1)]$
 $\quad \& \text{flg} [m(\text{idx}-1)]$

find-minimal-idx-1

min-idx

generate

min-flg[idx] = min-idx == 64
liter

find-minimal-idx-1 → bst-len

↓
bst-idx

↓
literal

flg

val-i

3. OUTPUT

excel 0

length 257-276 + 3

excel 2

→ code

excel 1

distance 0 -11 + 4

literal → excel 2 →

length → excel 0 + 3 → excel 2 → + 3

distance → excel 1 + 3 →

拼 32