InterpolationFilters[1]

In this documents a summary of the basic components implemented has been reported. This is not a complete reference guide for the architectures proposed, but a brief description of the hardware with pictures of the main blocks.

1-D DCT-IF architecture

In order to reduce the amount of energy per computation required by the 1-D architecture, a commonly adopted operand substitution method relies on replacing all the multiplications with additions and shifting operations. This is a particularly suitable technique with filter architectures since the multipliers coefficients are known at design time. However, keeping a certain order of coarse-grained reconfigurability in a multiplier-less approach is not easily achievable as with direct form FIR filters. Several papers have been proposed in order to find a reconfigurable multiplier-less 1-D filter architecture, the one chosen in this work of thesis is a slightly modified version of the filters introduced by Diniz et al. in [1]. In Table 3 the add and shift replacements with the respective multipliers coefficients have been reported.
Since the $1/4^{\text{th}}$ and the $3/4^{\text{th}}$ luma interpolation filters are symmetric, there is no need to design two different filter architectures, relying on a rotation of the input samples. The same is true for the chroma components, where only 4 of the 7 filters should be implemented while the others can rely on input rotations (Tabs. 1-2).

l	-3	-2	-1	0	1	2	3	4
hfilter$[l]$	-1	4	-11	40	40	-11	4	-1
qfilter$[l]$	-1	4	-10	58	17	-5	1	-

Table 1: Filter coefficients for luma fractional sample interpolation

l	-1	0	1	2
filter1$[l]$	-2	58	10	-2
filter2$[l]$	-4	54	16	-2
filter3$[l]$	-6	46	28	-4
filter4$[l]$	-4	36	36	-4

Table 2: Filter coefficients for Chroma fractional sample interpolation

In Figs. 1-2[2] the filters architectures implemented has been reported. Concerning the bit-depth of the intermediate tree stages, each adder has been designed considering the actual minimum number of bits required to represent correctly the data without overflow errors. An input configuration vector allows to choose the inputs to be processed given the filtering operation to be put in place.

coeff / shift	**1**	**2**	**4**	**5**	**6**	**10**	**11**	**16**	**17**	**28**	**36**	**40**	**46**	**54**	**58**
x	+			+			+		+						
$x \ll 1$		+			+	+	+						−	−	+
$x \ll 2$			+	+	+					−	+				
$x \ll 3$						+	+					+		−	−
$x \ll 4$								+	+				+		
$x \ll 5$										+	+	+	+		
$x \ll 6$														+	+

Table 3: Luma and chroma legacy coefficient multiplications replaced by add/shift

2-D DCT-IF legacy hardware architecture

Datapath

Several components form the datapath of the 2-D interpolation filter architecture proposed. Fig. 3 shows the implementation chosen for the luma hardware design:

- **Shift Registers Bank (SRB)**: a custom registers bank solution has been chosen as an input buffer, where each row corresponds to a different shift register. Precisely there are seven 8-bit output ports corresponding to the $N_{tap,\max} - 1$ inputs of the

[2] ***Note:*** in the DFGs reported the black bit-depth refers to the 1st stage interpolation filters with 8-bit input data, the blue one to the 2nd stage interpolation filters. The input-output bit-depth mismatches of the multiplexers are only present in the picture to get a shorthand notation. In the hardware implementation the input data width has been 0 padded for the MSBs of the 1st stage filters, and sign extended for the 2nd stage filters, matching the maximum input data width of the multiplexers. Where a three inputs mux is shown, the missing selection bits select the same input as one of the other defined combinations.

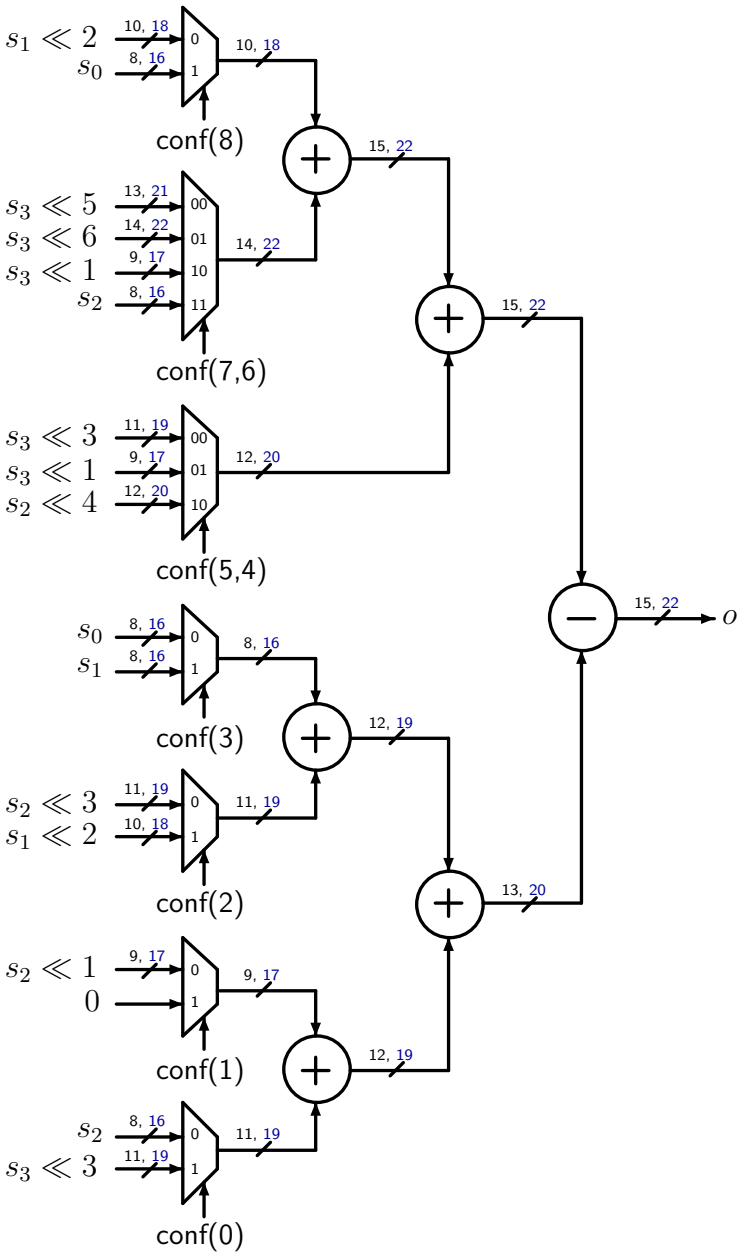

Figure 1: Reconfigurable legacy luma half filter[2]

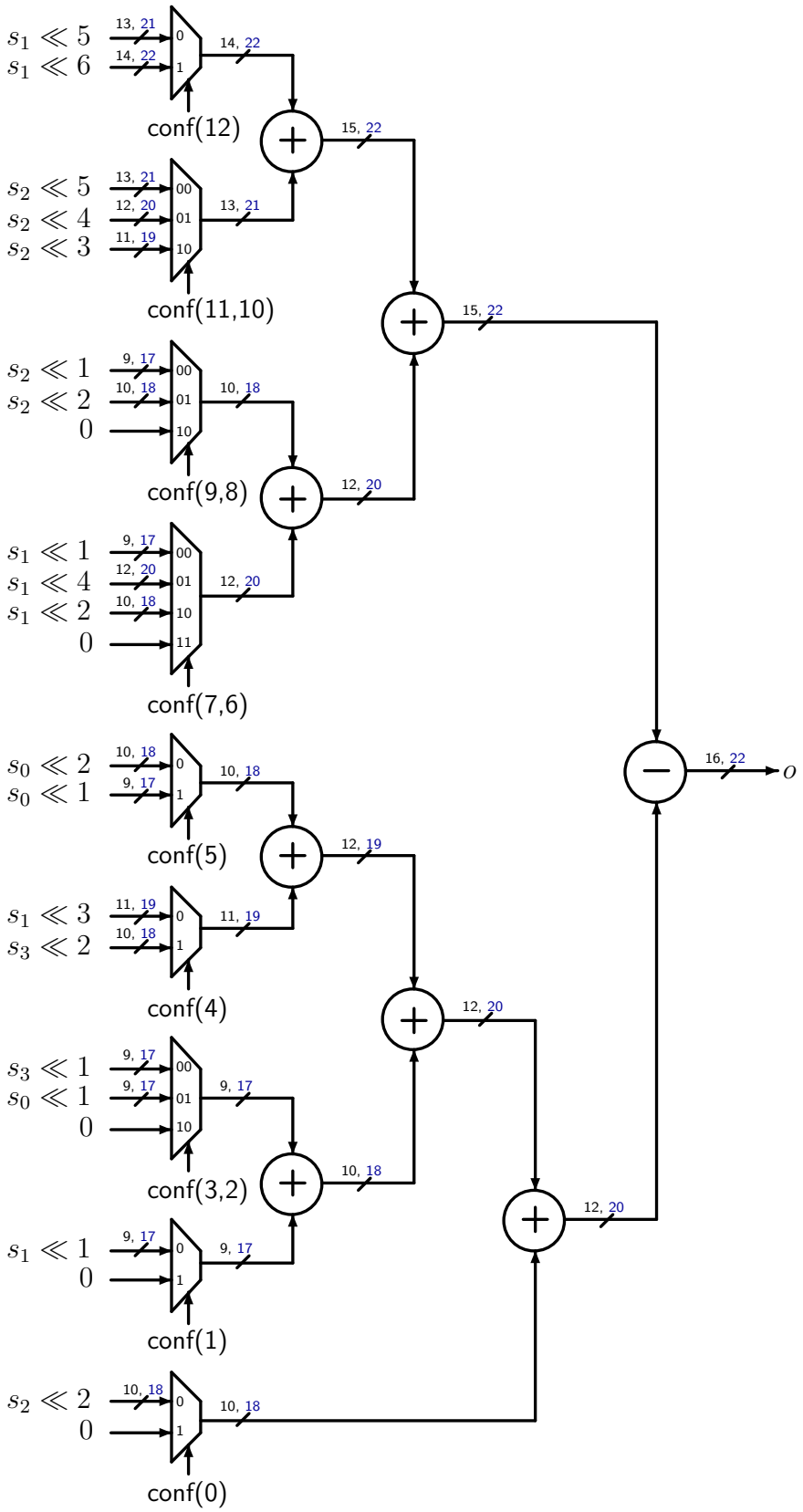

Figure 2: Reconfigurable legacy chroma filter[2]

luma reconfigurable interpolation filter. The address counter select which row[3] (7 bytes) of the memory has to be provided to the routing unit input, as well as what first row sample should be updated with a pixel coming from the external user. Every shift register shifts its content only when it is pointed by the address counter. This solution has several advantages in term of power consumption when clock gating is considered.

- **Address Counter (CNT)**: a programmable counter is used to point to a shift register in the SRB. It is used to fill the lines required by the architecture to start the filtering process. The counter modulus is provided as an input of the datapath coming from the external user and it must be equal to:
  - $H + N_{tap} - 2$, where H is the height of the prediction block to be processed by a 2-D filtering operation[4].
  - $W-1$ or $H-1$ where W is the width and H is the height of the prediction block to be processed by a 1-D vertical or horizontal filtering operation respectively[4].
- **Routing Unit (RtU)**: because of the 1-D filter reconfigurable structure (Fig. 1), a routing unit is needed to guide the outputs of the memory bank to the correct inputs of the filter. The same is true for the second stage filter, where data are coming from a shift register.
- **DCT-IF**: two 1-D luma filters (Fig. 1) have been inserted in the datapath.
- **Shift Register (SR)**: a Serial Input Parallel Output (SIPO) 8x16 bits shift register is required as spatial delay line of the second 1-D filter, to temporarily save data coming from the 1st stage filter.
- **Rounding Unit**: HEVC implements a round half up operation adding half quantization step and then truncating either 12 bits if the data come from a complete 2-D filtering operation, or 6 bits if the data come from a 1-D filtering operation.
- **Clipping Unit**: since the filter output is signed and represented with more then 8-bit, a clipping unit is required working with saturation arithmetic. The output are clipped to 255 if positive and to 0 if negative, to retrieve the 8-bit unsigned data format required by the standard.

All the registers included in the datapath, as well as the SRB, the CNT and the SR have an asynchronous reset input, not shown in the figure.
As can be seen in Fig. 3, a set of multiplexers determine which output should be provided. The architectures proposed work with both the Motion Estimation and the Motion Compensation tool of HEVC. In MC the filter output bit-depth is different from the 8-bit unsigned data format when bi-prediction is involved. In this particular case when only a 1-D filtering operation is required, the output is not rounded and/or clipped and no truncation is performed. On the other hand, when bi-prediction is required with a 2-D

[3]Since the address input can potentially points to locations beyond the address space of the bank, a safe solution has been taken into account forcing the address input of the memory to a known location when the input address port exceed the memory dimension.

[4]The memory locations start from 0 and not from 1, hence an additional -1 was considered here.

complete filtering operation, the output is not rounded and/or clipped and the six right-most bits are truncated[5].
The chroma datapath architecture is almost equal to the luma one, with some changes related to the reduced maximum number of taps required by the chroma filters, like the input buffer dimension, the routing units and the 1-D filters.

Control Unit

The control machine of the datapath is composed by two Moore Finite State Machines (FSM) sharing a programmable counter, as reported in Fig. 4:

- **FSM1**: it controls the first stage filter branch setting the necessary configuration vectors required by the routing unit and the DCT-IF (8-bit input) to handle either a 1-D filtering operation or a 2-D one. It is also responsible for the output setting whenever a one dimensional interpolation is requested by the user. It starts the second FSM if a 2-D interpolation is required.

- **FSM2**: it controls the first stage filter branch setting the necessary configuration vectors required by the routing unit and the DCT-IF (16-bit input) to handle a 2-D filtering operation. It sets the output register when it is operating.

- **Shared Counter (SCNT)**: a shared programmable counter is used by both the FSMs:

  - FSM1 resorts to this component to count how many lines are filled in the SRB (Fig. 3). It is needed since, depending on the number of tap required, 7 or 8 in the luma legacy case, the input buffer line filled with input samples must be 6 or 7 respectively.
  - FSM2 adopts a programmable counter to control how many partly interpolated pixel are present in the second stage shift register. Because of data dependencies between the first stage interpolation filter and the second one, a stall is required when the last pixel of a column is reached. Thus, depending on the filter tap required, the second FSM stall the output for $N_{tap} - 1$ clock cycles, waiting for the SR to be filled with new samples.

Since the lines count needed by the first FSM is never simultaneous to the pixels count adopted by the second FSM, a shared counter solution has been chosen, instead of two separate ones. Multiplexers are used to handle the programmable counter sharing.
The LE of the output register is handled by the first FSM for a 1-D filtering case, by the second FSM for a 2-D filtering one.
The chroma control unit is equal to the luma one, thus it was not reported in the figures. Also the two Moore FSMs state transition graphs are quite similar. Since the luma filter have 4-tap instead of the luma 7-tap or 8-tap, and 7 filters can be implemented instead of 3 by a single reconfigurable DCT-IF, different set states have been considered following the IDLE state in the STGs.

In the following a brief description of the FSMs states has been reported:

[5]This operation does not affect the data precision since the truncated bits would not have been involved in future rounding steps.

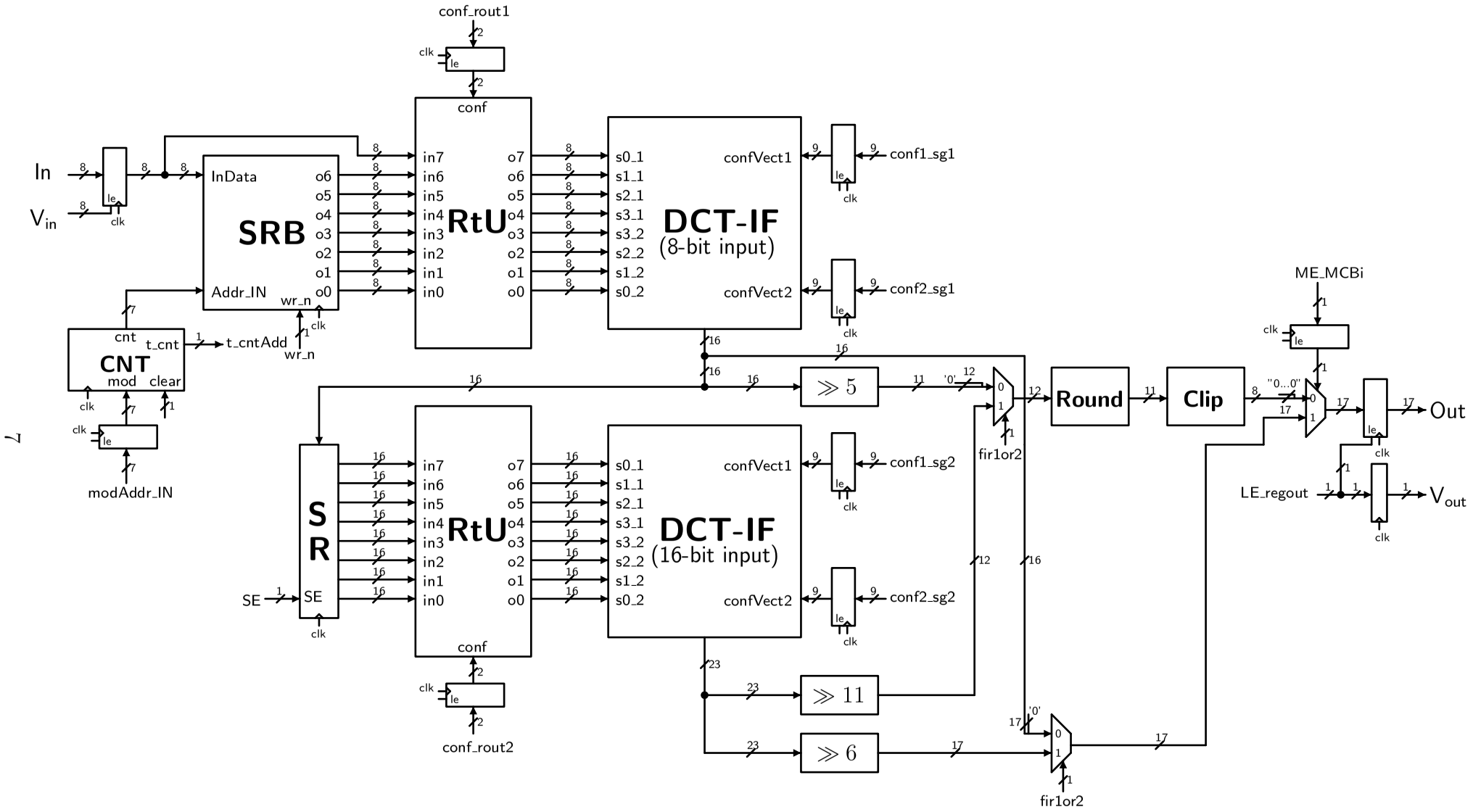

Figure 3: Datapath 2-D DCT-IF luma legacy

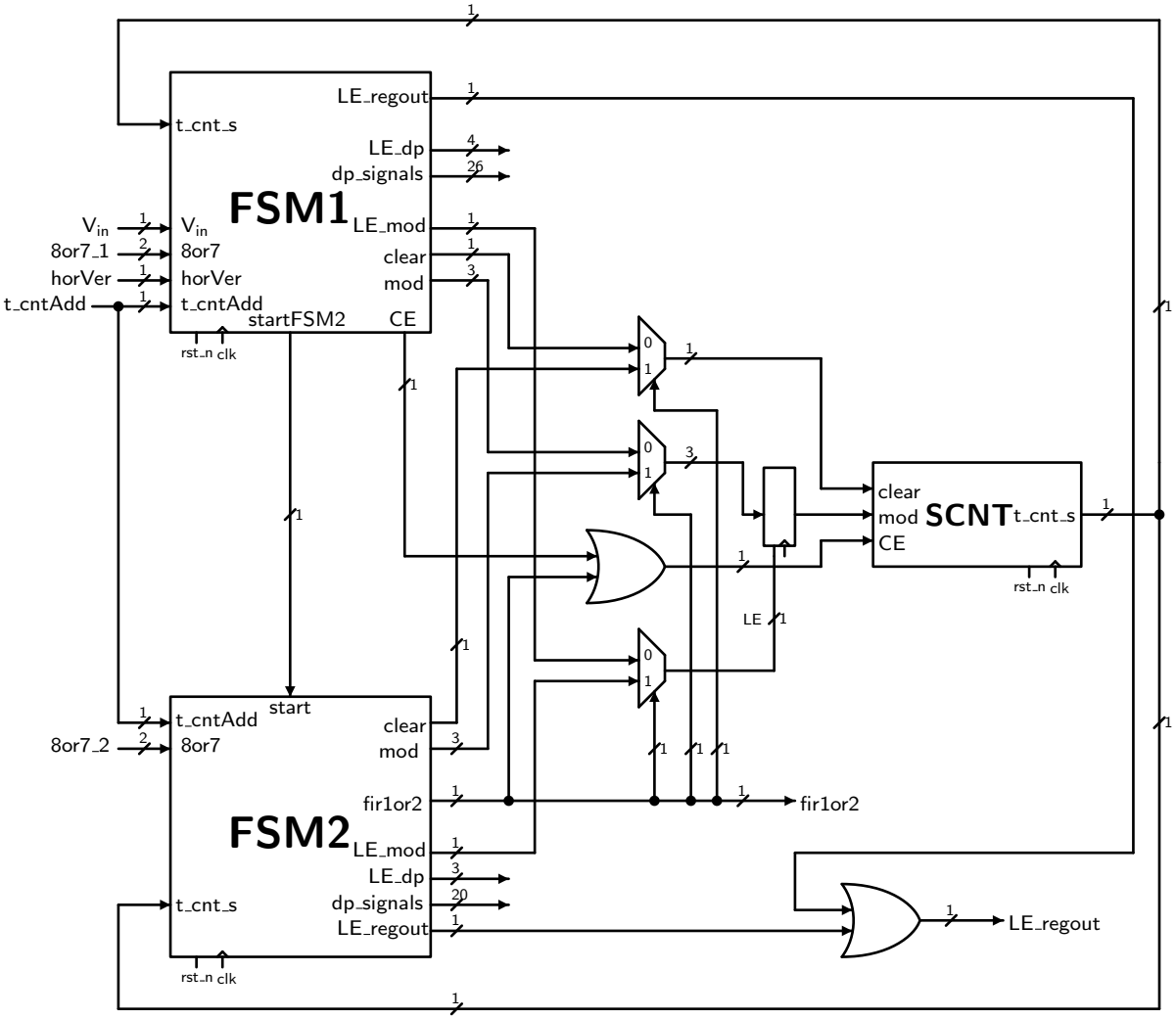

Figure 4: Control Unit 2-D DCT-IF

- FSM1:
  - **IDLE**: wait for a start condition coming from the user.
  - **SET71**, **SET72**, **SET8**: sets the routing unit and the DCT-IF configuration registers of the first stage branch.
  - **WAIT_L**: wait for a line to be written in the input SRB buffer.
  - **LINE_1**: increment the line counter (SCNT).
  - **SETOUT**: start setting the output register. If a 1-D filtering is required then the FSM1 is responsible for the output setting.
  - **START**: start the second FSM since a 2-D filtering operation is required.
- FSM2:
  - **IDLE**: wait for a start condition coming from the FSM1.
  - **SET71**, **SET72**, **SET8**: sets the routing unit and the DCT-IF configuration registers of the second stage branch.
  - **WAIT_F**: wait the shift register (SR) to be full with $N_{tap} - 1$ data.
  - **WAIT_L**: start setting the output and wait for a line to be written in the input SRB buffer.
  - **LAST7**, **LAST8**: sample the last pixel before stalling the output, reset the shared counter. This states have been introduced to solve timing issues with the address terminal count.

Processing Element

The processing element (PE) luma top level (Fig. 7) is composed by the datapath and by the control unit presented in Fig. 3-4 respectively. Concerning the input-output protocol, input samples must be provided at the same time as the required configuration signals (Figs. 8-9). The PE provides a sampling strobe (V_{out}) synchronized with the output data. A summary of the input signals is reported in the following:

- $\mathbf{V_{in}}$: sampling strobe used by the PE to store the input pixels and to start the filtering process.
- **InData**: 8-bit unsigned pixels input.
- **horVer**: if '0' the hardware is programmed for a 1-D filtering operation, if '1' it allows for a 2-D filtering operation.
- **8or7_1**, **8or7_2**: set which interpolation filter must be selected for the first stage DCT-IF branch and for the second one respectively[6]:
  - "00": 8-tap half-pel filter.

[6] ***Note***: even if the "01" 8or7 case is not reported since useless for the filter selection, it is covered for safety reasons. In the state transition graphs if the "01" selection is detected both the FSMs come back to the IDLE state.

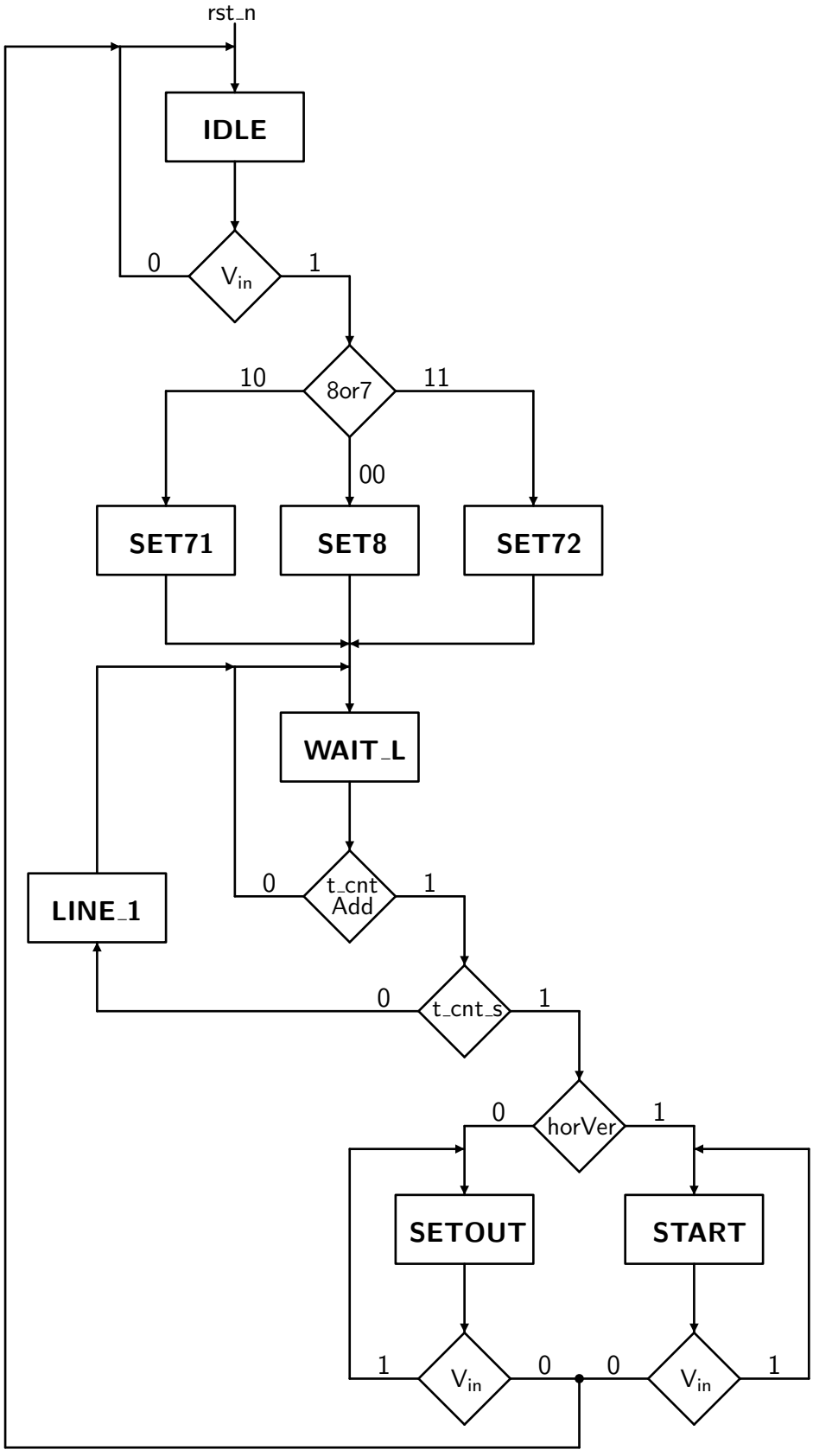

Figure 5: State transition graph luma legacy FSM1[6]

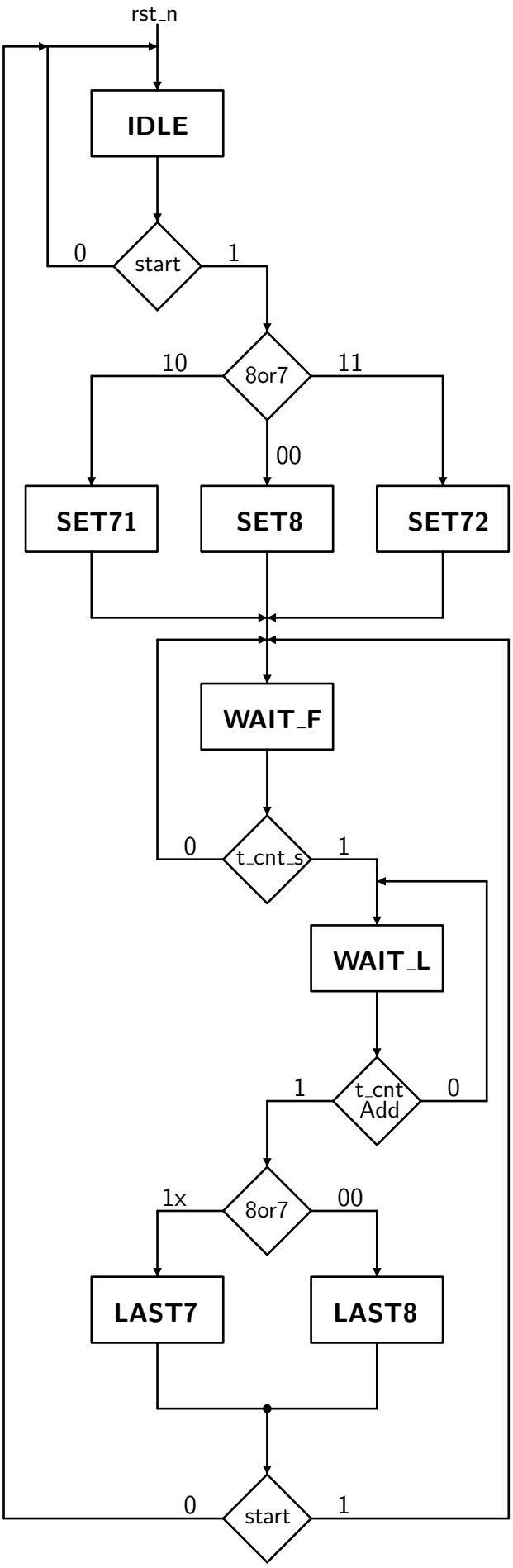

Figure 6: State transition graph luma legacy FSM2[6]

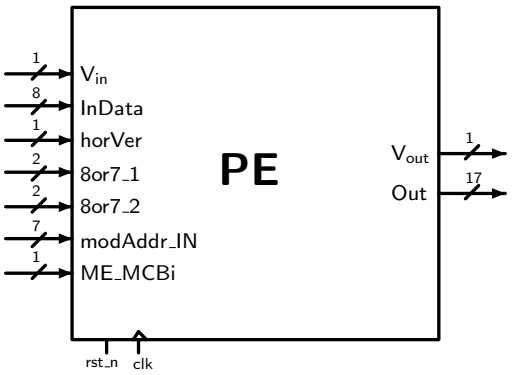

Figure 7: Top level processing element

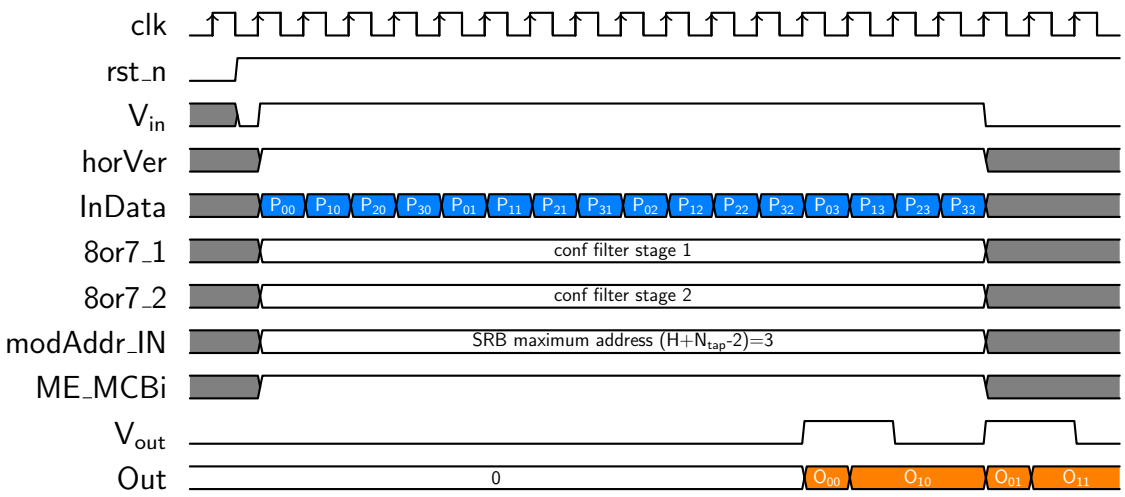

Figure 8: 2-D filtering example: 2x2 dummy pixels block, 3 tap filters, with bidirectional prediction output

  - "10": 7-tap $1/4^{\text{th}}$ quarter-pel filter.
  - "11": 7-tap $3/4^{\text{th}}$ quarter-pel filter.

- **modAddr_IN**: input modulus of the address programmable counter. This must be set accordingly to what reported in Sec. to the CNT entry in the description.

- **ME_MCBi**: if '0' the output is provided as 8-bit unsigned, as the input format. This is required by both the Motion Estimation and the Motion Compensation tool. If '1' the output is kept to higher precision, not affected by rounding and/or clipping operations, as required by the MC bi-prediction case.

2-D DCT-IF approximate hardware architecture

Providing power efficient designs to cope with the compelling demand for long battery life is still a challenging task. Approximate computing can further reduce the energy and the power consumption of the system, offering energy versus quality tradeoff.

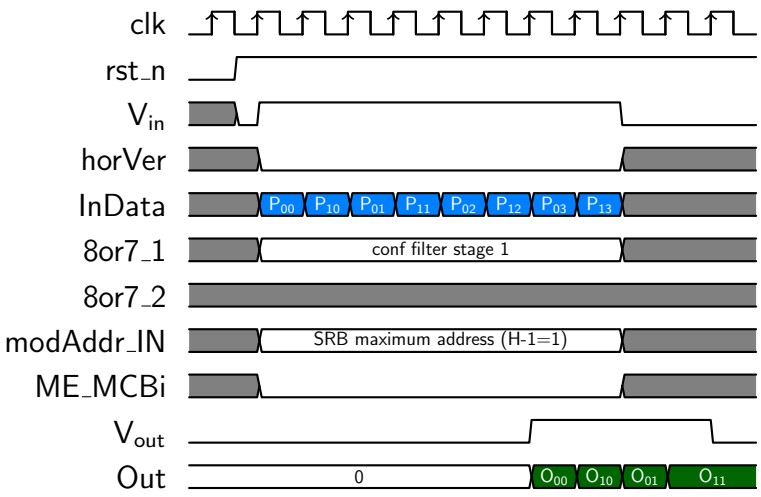

Figure 9: 1-D horizontal filtering example: 2x2 dummy pixels block, 3 tap filters, with no bidirectional prediction output

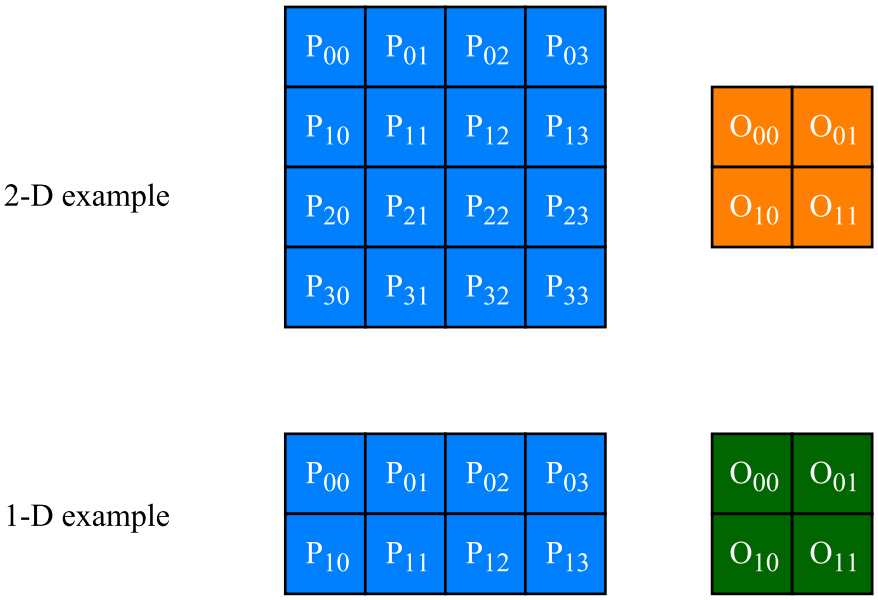

Figure 10: Input and output pixels of the examples in Fig. 8-9

In the following a reconfigurable architecture is proposed, exploiting the approximation trends reported in [2]. The 3-tap and 5-tap configurations were considered for the luma datapath, the 2-tap one for the chroma datapath (Tab. 4-5).

N_{tap}	$\alpha = 1/4$	$\alpha = 1/2$
Legacy	$-1, 4, -10, 58, 17, -5, 1$	$-1, 4, -11, 40, 40, -11, 4, -1$
7	$-1, 4, -10, 58, 17, -5, 1$	$-1, 4, -11, 40, 40, -11, 3$
5	$1, -6, 20, 54, -5$	$2, -9, 40, 40, -9$
3	$-4, 20, 48$	$-9, 41, 32$
1	64	64

Table 4: Luma filter coefficients [2]

α	**Legacy**	$N_{tap} = 1$	$N_{tap} = 2$	$N_{tap} = 3$
1/8	$-2, 58, 10, -2$	64	$57, 7$	$-3, 62, 5$
2/8	$-4, 54, 16, -2$	64	$50, 14$	$-5, 58, 11$
3/8	$-6, 46, 28, -4$	64	$41, 23$	$-7, 51, 20$
4/8	$-4, 36, 36, -4$	64	$32, 32$	$-6, 42, 28$

Table 5: Chroma filter coefficients [2]

Higher order luma filters were not considered because the energy benefit results reported in [3] do not encourage such a choice since it gave no energy consumption reduction with respect to the legacy implementation[7].
Concerning the 1-D filter implementation, the same multiplier-less approach has been followed, as the legacy design. Figs. 11-12 report the 5-tap and the 3-tap reconfigurable luma DCT-IFs implementations respectively, in Fig. 13 the reconfigurable chroma 2-tap architecture is shown.

coeff / **shift**	**1**	**2**	**4**	**5**	**6**	**7**	**9**	**14**	**20**	**23**	**32**	**40**	**41**	**48**	**50**	**54**	**57**
x	+			+		−	+			−			+				+
$x \ll 1$		+			+			−							+	+	
$x \ll 2$			+	+	+				+							+	
$x \ll 3$						+	+			−		+	+				−
$x \ll 4$								+	+					+	+	+	
$x \ll 5$										+	+	+	+	+	+	+	
$x \ll 6$																	+

Table 6: Luma and chroma approximate coefficient multiplications replaced by add/shift

[7]Since the architecture here proposed is quite different from the one in the paper, it could be useful to try with higher order solutions in future.

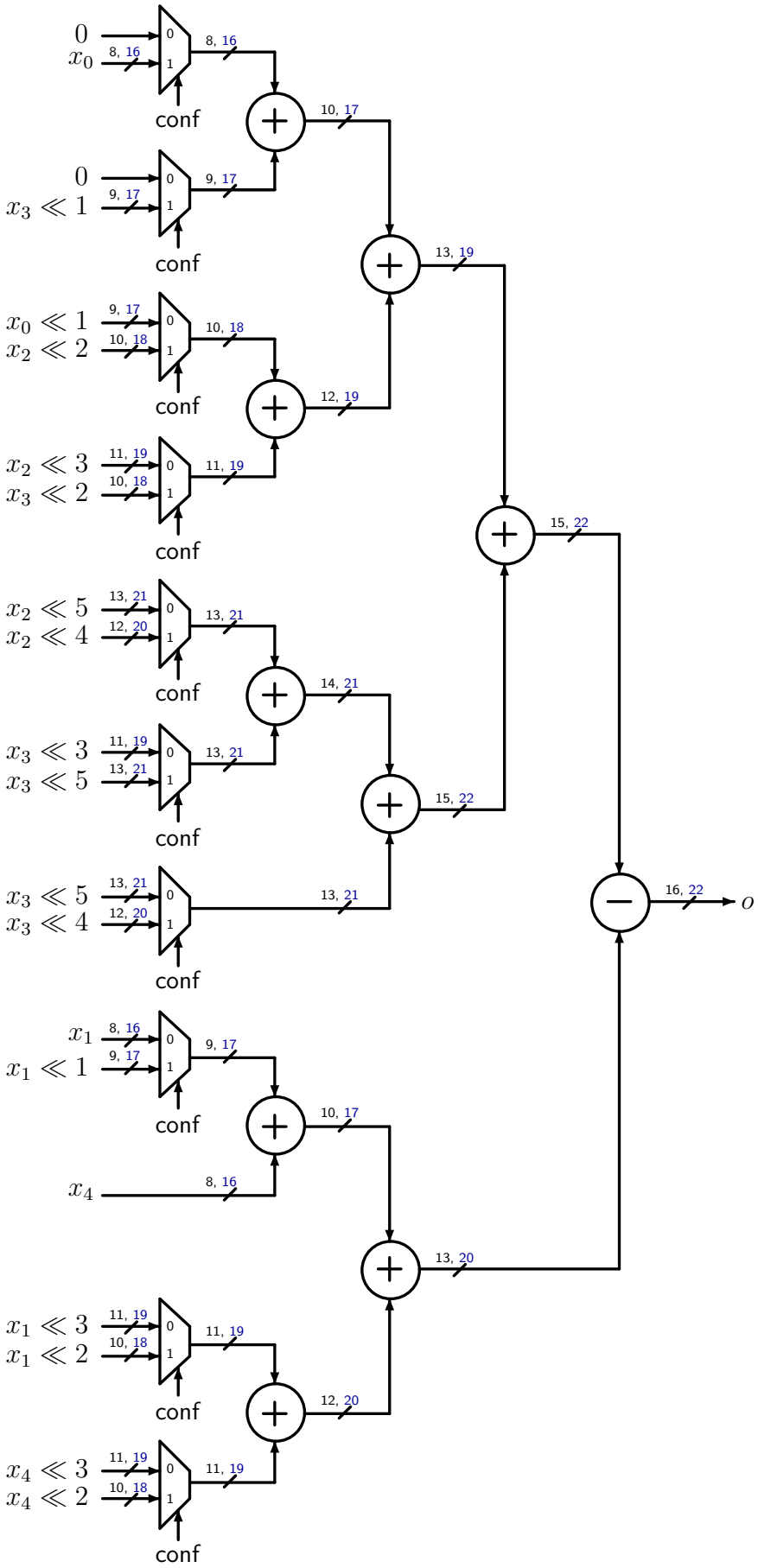

Figure 11: Reconfigurable approximate luma 5-tap filter[2]

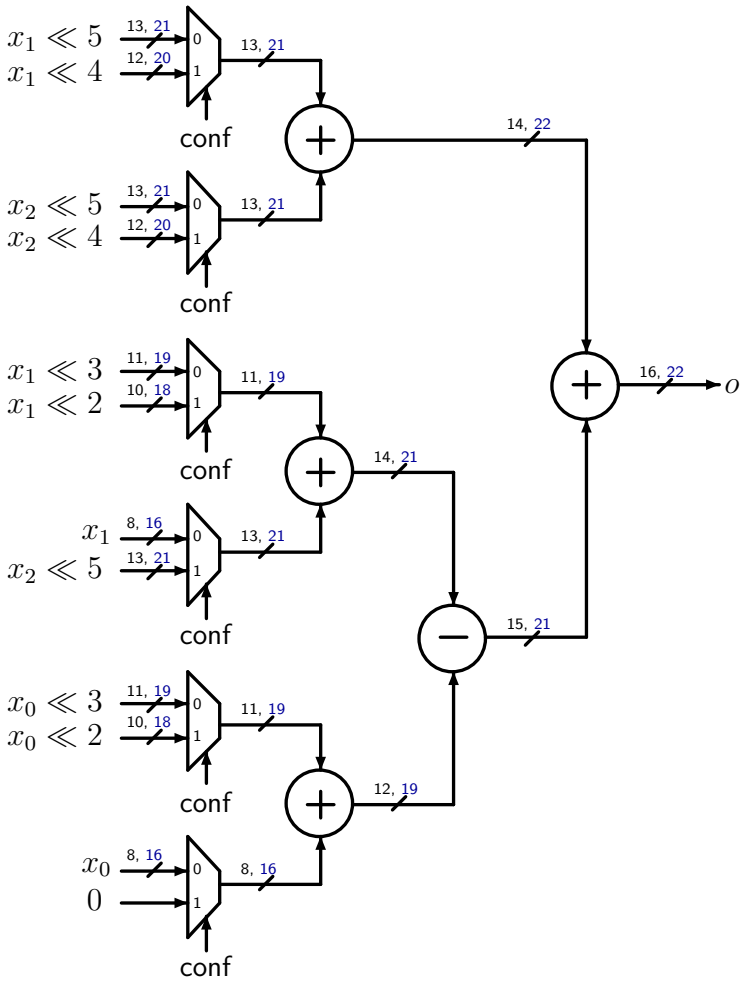

Figure 12: Reconfigurable approximate luma 3-tap filter[2]

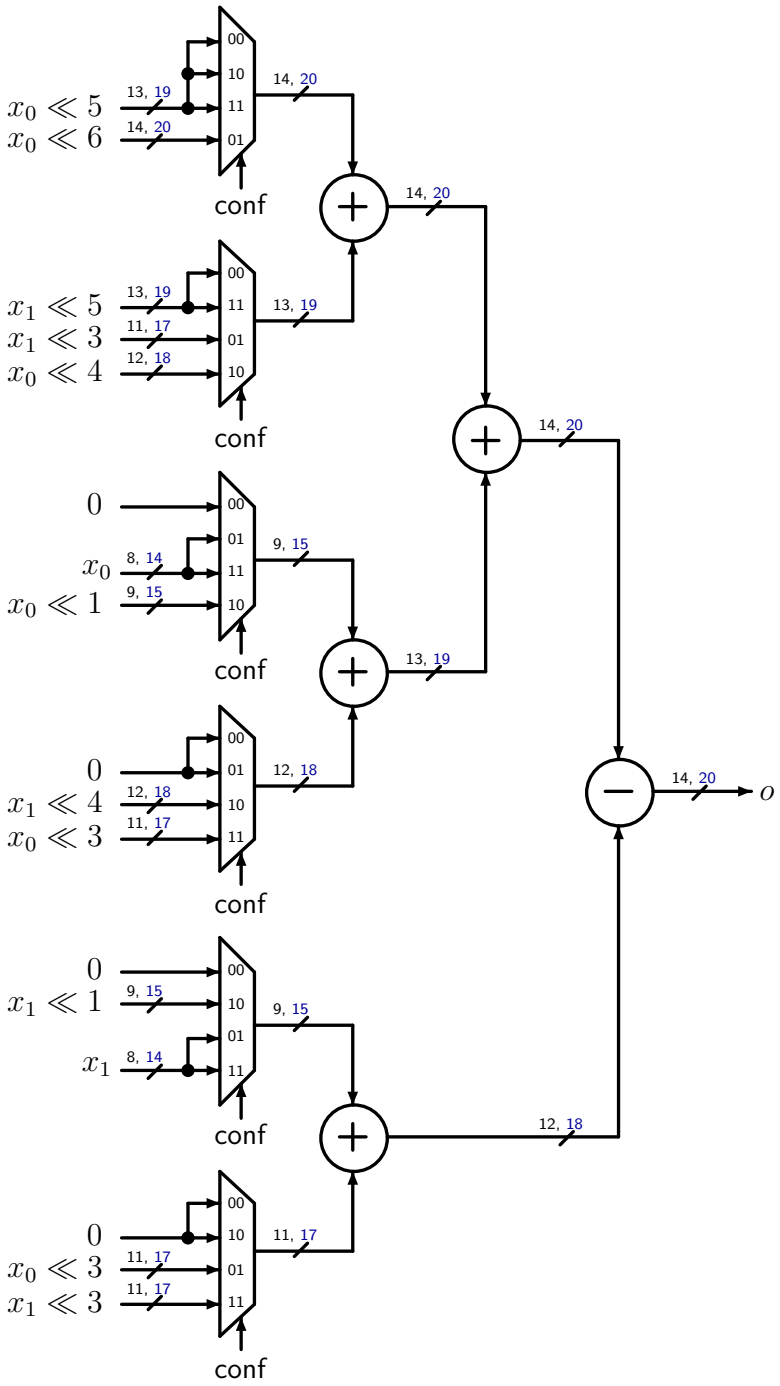

Figure 13: Reconfigurable approximate chroma 2-tap filter[2]

The datapath is composed by different parallel filter branches, each one related to a specific reconfigurable DCT-IF implementation (Fig. 14). Depending on the input requirements, muxes select which branch output should be considered for the first and for the second stage. The routing units set the DCT-IF inputs to zero if the referring filter is not used during the computation[8]. This solution is useful to reduce the power consumption of the design. Placing a battery of AND ports to the routing units outputs prevent the switching of the filter inputs, reducing the total activity. Both the FSMs are almost unchanged from the previous ones. More set states and a wider filter selection port have been considered because of the larger filtering options, nevertheless the overall structure is the same as the previous one. Also the top level entity is almost the same as the one reported in Fig. 7, with small changes in the bit-depth of the 8or7 signals, that now are used to select among a larger number of filters (their name has been changed to filterSel_1 and filterSel_2).

[8]Instead of adding demultiplexers to the routing unit inputs, the blocking behaviour has been embedded in the routing units themselves considering the not covered combination of the configuration vector.

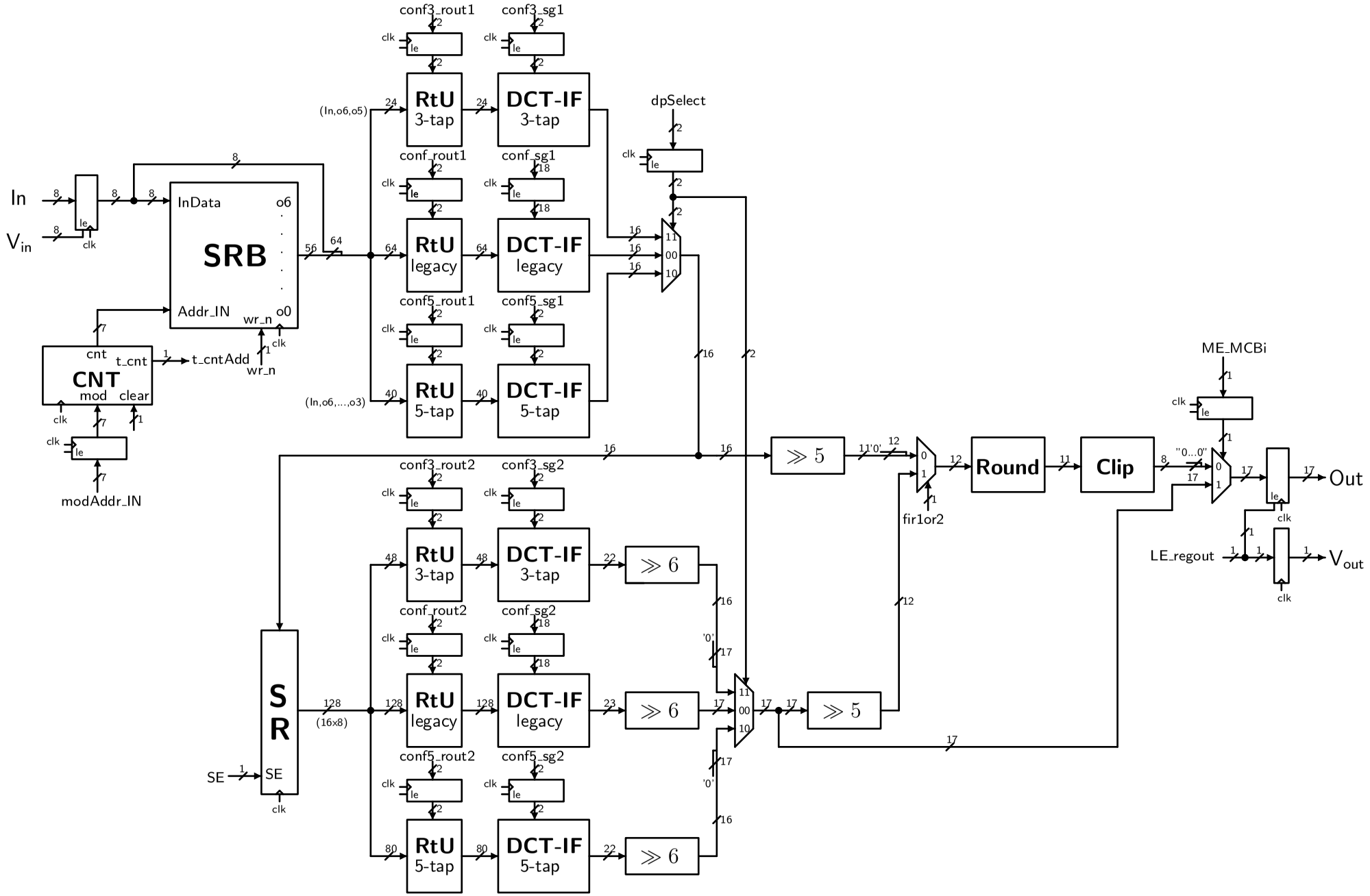

Figure 14: Datapath 2-D DCT-IF luma approximate

Bibliography

[1] C. M. Diniz, M. Shafique, S. Bampi, and J. Henkel, "A reconfigurable hardware architecture for fractional pixel interpolation in high efficiency video coding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 238–251, Feb 2015.

[2] E. Nogues, D. Menard, and M. Pelcat, "Algorithmic-level approximate computing applied to energy efficient hevc decoding," *IEEE Transactions on Emerging Topics in Computing*, vol. PP, no. 99, pp. 1–12, 2016.

[3] C. Sau, F. Palumbo, M. Pelcat, J. Heulot, E. Nogues, D. Menard, P. Meloni, and L. Raffo, "Challenging the best hevc fractional pixel fpga interpolators with reconfigurable and multifrequency approximate computing," *IEEE Embedded Systems Letters*, vol. 9, pp. 65–68, Sept 2017.