

buffering_period(payloadSize) {	Descriptor
seq_parameter_set_id	ue(v)
if(!sub_pic_cpb_params_present_flag)	
rap_cpb_params_present_flag	u(1)
if(NalHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
}	

FIG 1

pic_timing(payloadSize) {	Descriptor
if(CpbDpbDelaysPresentFlag) {	
cpb_removal_delay	u(v)
dpb_output_delay	u(v)
if(sub_pic_cpb_params_present_flag) {	
num_decoding_units_minus1	ue(v)
for(i = 0; i <= num_decoding_units_minus1; i++) {	
num_nalus_in_du_minus1[i]	ue(v)
du_cpb_removal_delay[i]	u(v)
}	
}	
}	
}	

FIG 2

vui_parameters() {	Descriptor
aspect_ratio_info_present_flag	u(1)
if(aspect_ratio_info_present_flag) {	
aspect_ratio_idc	u(8)
if(aspect_ratio_idc == Extended_SAR) {	
sar_width	u(16)
sar_height	u(16)
}	
}	
overscan_info_present_flag	u(1)
if(overscan_info_present_flag)	
overscan_appropriate_flag	u(1)
video_signal_type_present_flag	u(1)
if(video_signal_type_present_flag) {	
video_format	u(3)
video_full_range_flag	u(1)
colour_description_present_flag	u(1)
if(colour_description_present_flag) {	
colour_primaries	u(8)
transfer_characteristics	u(8)
matrix_coefficients	u(8)
}	
}	
chroma_loc_info_present_flag	u(1)
if(chroma_loc_info_present_flag) {	
chroma_sample_loc_type_top_field	ue(v)
chroma_sample_loc_type_bottom_field	ue(v)
}	
neutral_chroma_indication_flag	u(1)

FIG 3A

FIG 3

FIG 3A

FIG 3B

field_seq_flag	u(1)
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
fixed_pic_rate_flag	u(1)
}	
nal_hrd_parameters_present_flag	u(1)
if(nal_hrd_parameters_present_flag)	
hrd_parameters()	
vcl_hrd_parameters_present_flag	u(1)
if(vcl_hrd_parameters_present_flag)	
hrd_parameters()	
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag)	
low_delay_hrd_flag	u(1)
sub_pic_cpb_params_present_flag	u(1)
if(sub_pic_cpb_params_present_flag)	
num_units_in_sub_tick	u(32)
}	
bitstream_restriction_flag	u(1)
if(bitstream_restriction_flag) {	
tiles_fixed_structure_flag	u(1)
motion_vectors_over_pic_boundaries_flag	u(1)
max_bytes_per_pic_denom	ue(v)
max_bits_per_minicu_denom	ue(v)
log2_max_mv_length_horizontal	ue(v)
log2_max_mv_length_vertical	ue(v)
}	
}	

FIG 3B

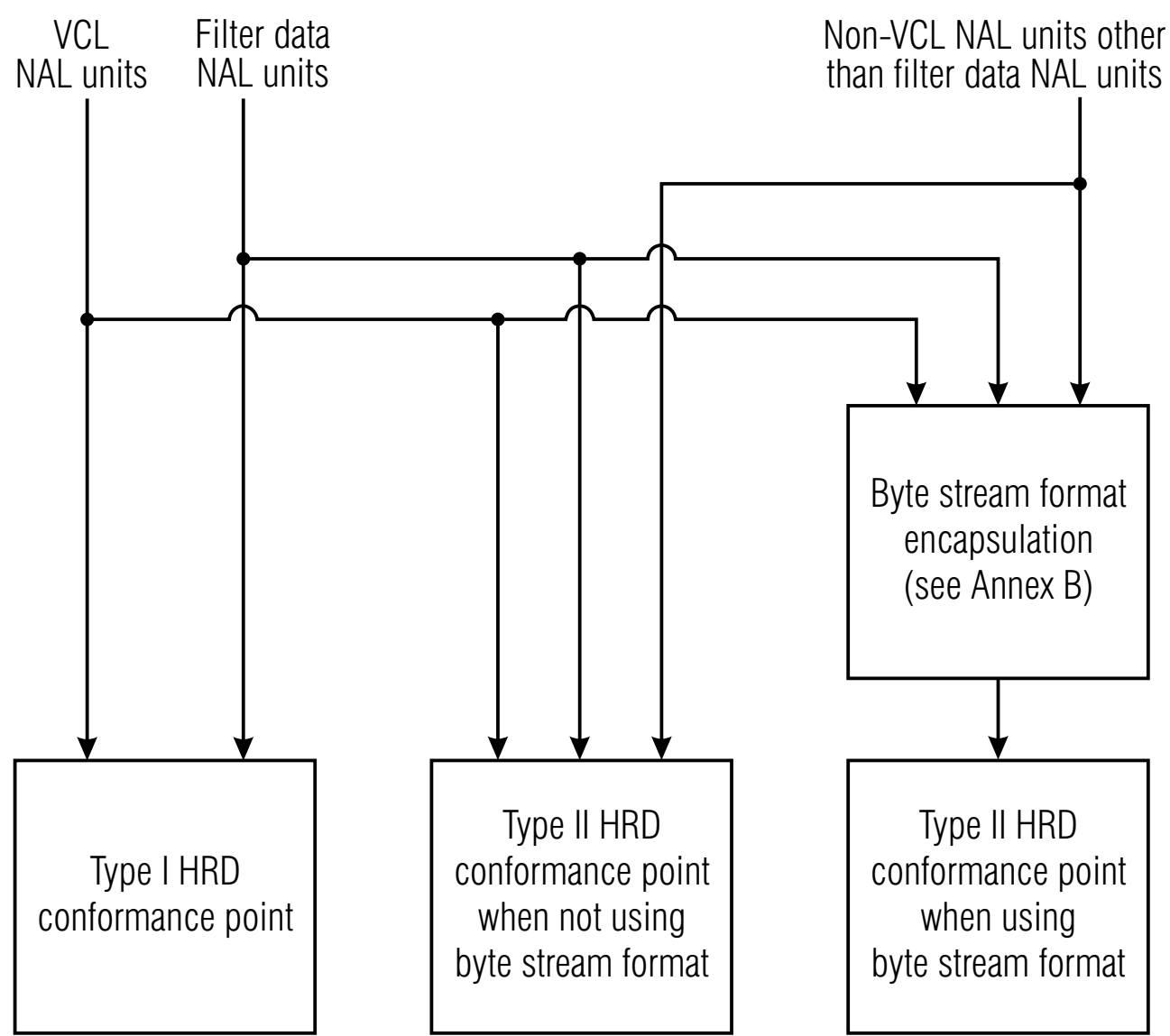
FIG 3	FIG 3A
	FIG 3B

hrd_parameters() {	Descriptor
cpb_cnt_minus1	ue(v)
bit_rate_scale	u(4)
cpb_size_scale	u(4)
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
bit_rate_value_minus1[SchedSelIdx]	ue(v)
cpb_size_value_minus1[SchedSelIdx]	ue(v)
cbr_flag[SchedSelIdx]	u(1)
}	
initial_cpb_removal_delay_length_minus1	u(5)
cpb_removal_delay_length_minus1	u(5)
dpb_output_delay_length_minus1	u(5)
time_offset_length	u(5)
}	

FIG 4

filler_data_rbsp() {	Descriptor
while(next_bits(8) == 0xFF)	
ff_byte /* equal to 0xFF */	f(8)
rbp_trailing_bits()	
}	

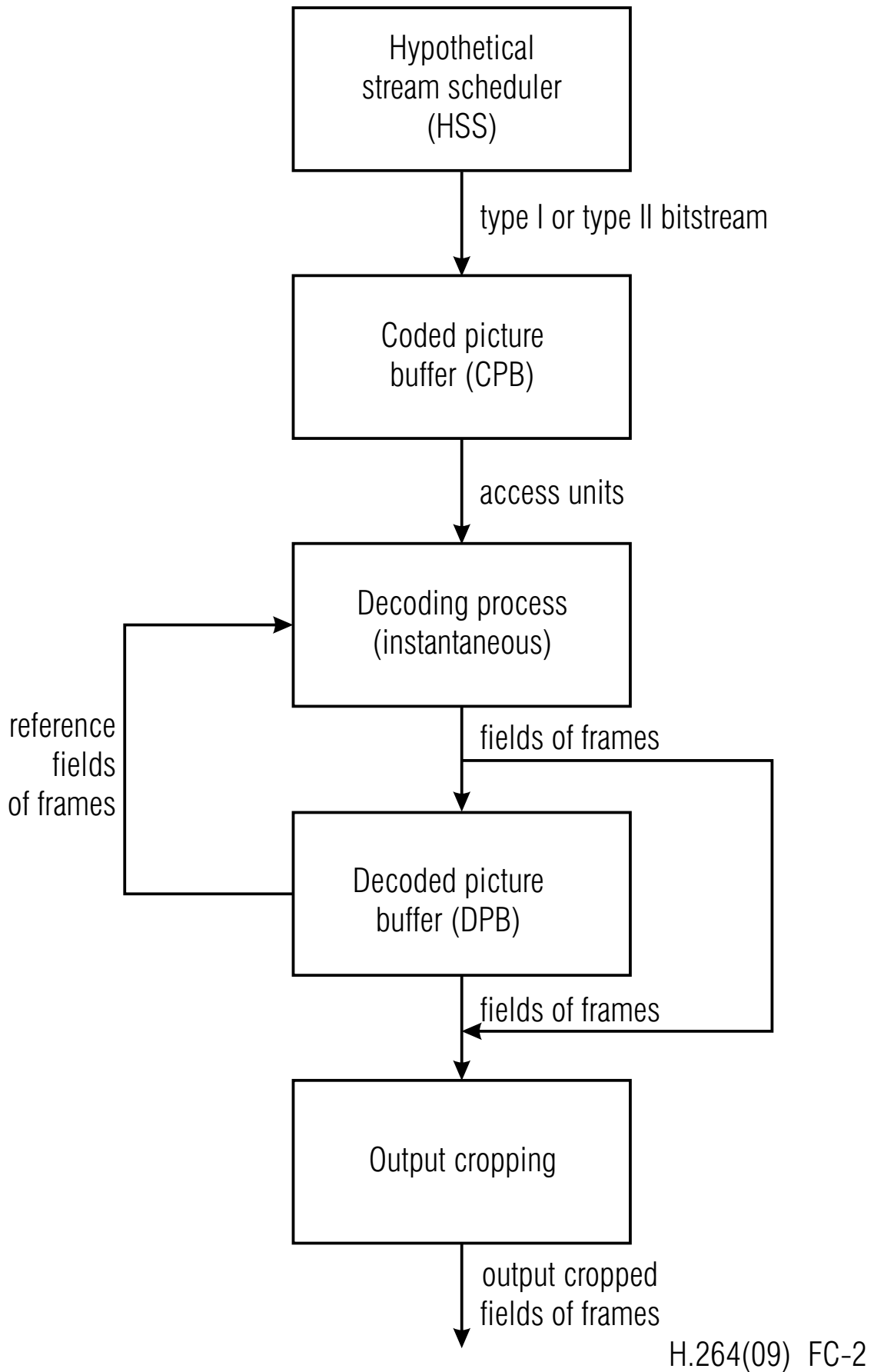
FIG 5



H.264(09)_FC-1

FIG 6

7/37



H.264(09)_FC-2

FIG 7

slice_header() {	Descriptor
first_slice_in_pic_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(!dependent_slice_flag) {	
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(RapPicFlag) {	
rap_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
if(!IdrPicFlag) {	
....	
}	
if(tiles_or_entropy_coding_sync_idc == 1 tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte	u(8)
}	
byte_alignment()	
}	

FIG 8

	Descriptor
pic_parameter_set_rbsp() {	
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
sign_data_hiding_flag	u(1)
if(sign_data_hiding_flag)	
sign_hiding_threshold	u(4)
cabac_init_present_flag	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
[Ed. (BB): not present in HM software]	
pic_init_qp_minus26	se(v)
constrained_intra_pred_flag	u(1)
slice_granularity	u(2)
diff_cu_qp_delta_depth	ue(v)
cb_qp_offset	se(v)
cr_qp_offset	se(v)
weighted_pred_flag	u(1)
weighted_bipred_idc	u(2)
output_flag_present_flag	u(1)
transquant_bypass_enable_flag	u(1)
dependent_slice_enabled_flag	u(1)
tiles_or_entropy_coding_sync_idc	u(2)
if(tiles_or_entropy_coding_sync_idc == 1){	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
} else if(tiles_or_entropy_coding_sync_idc == 3)	
cabac_independent_flag	u(1)
...	
}	

FIG 9

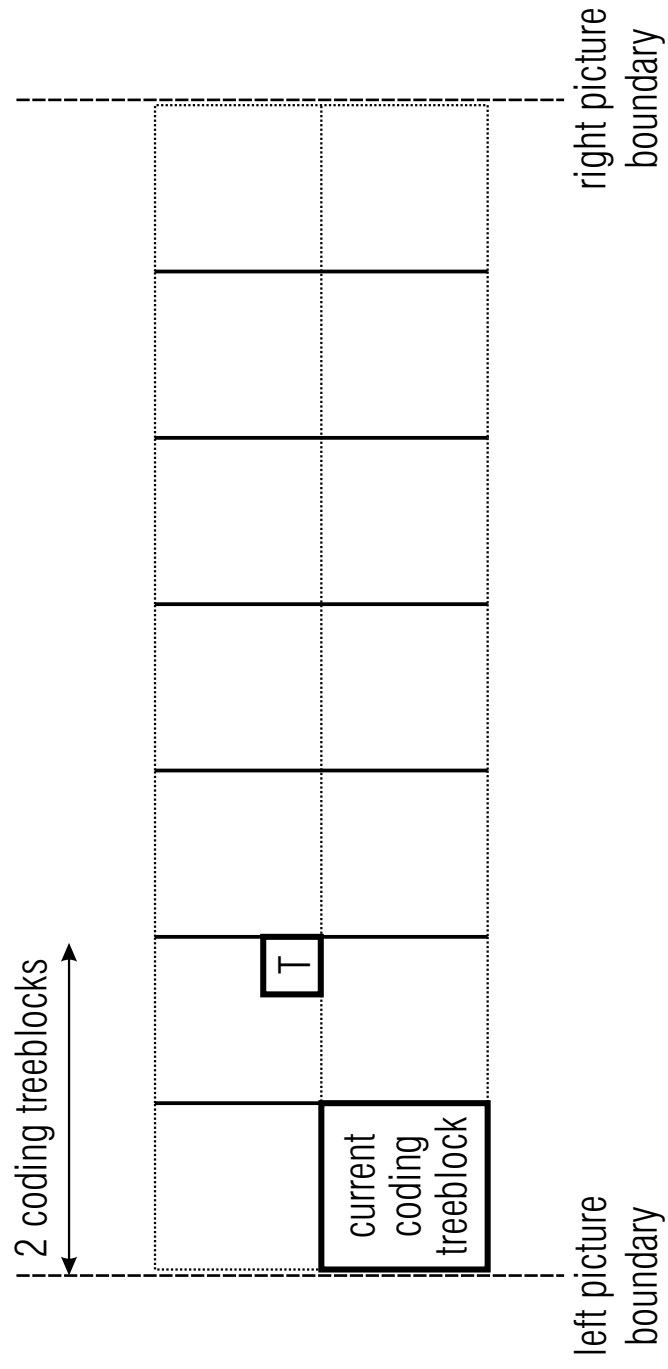


FIG 10A

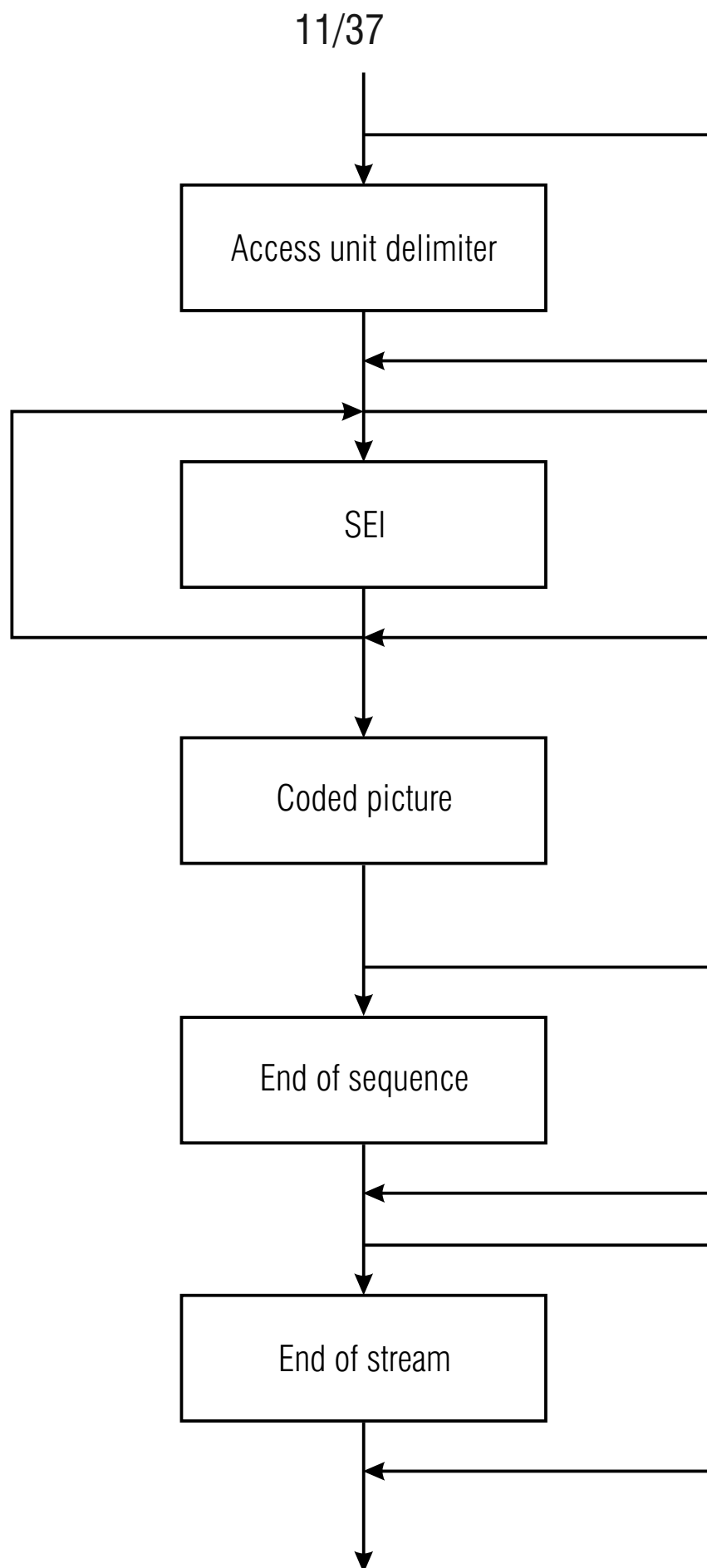


FIG 10B

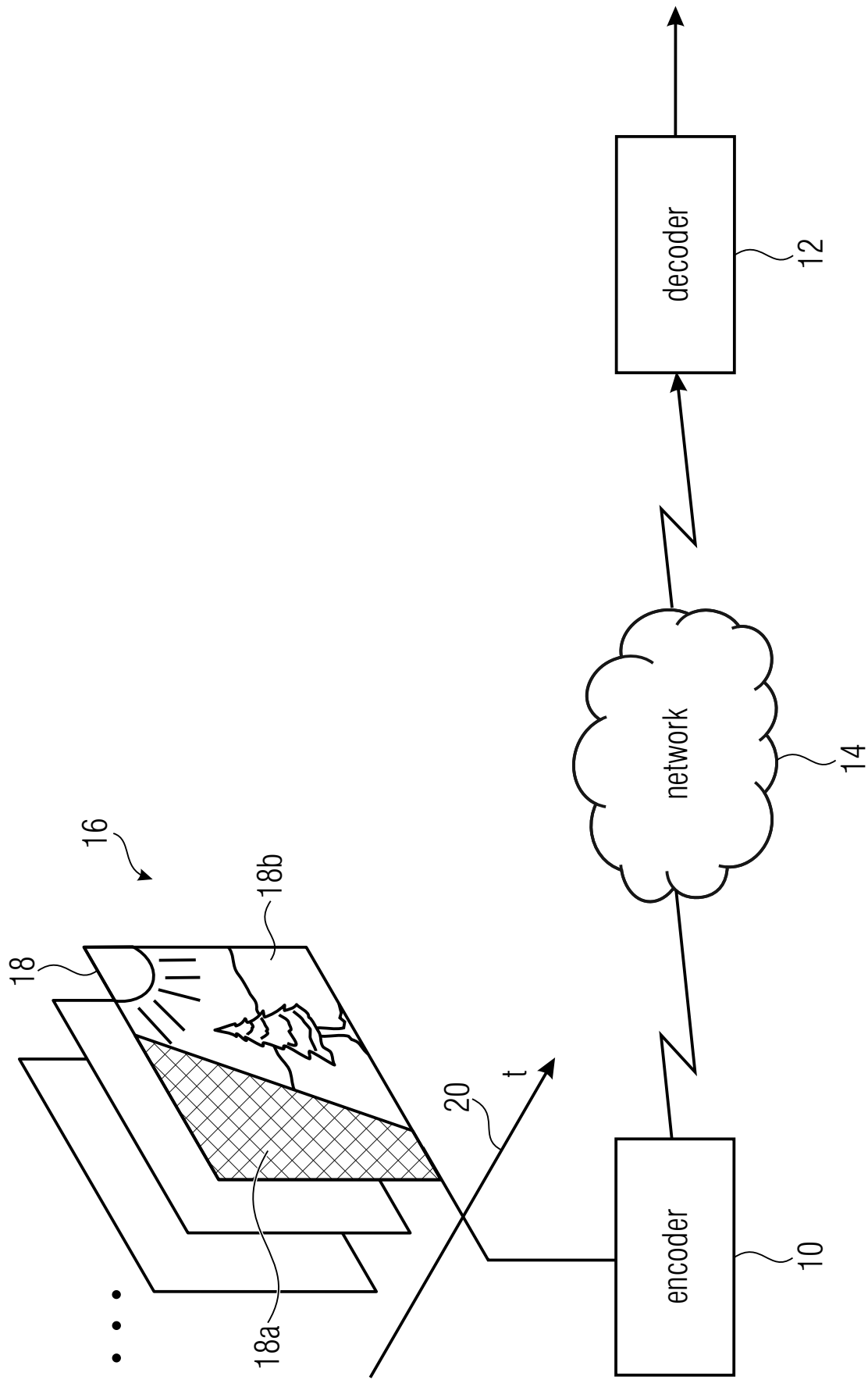
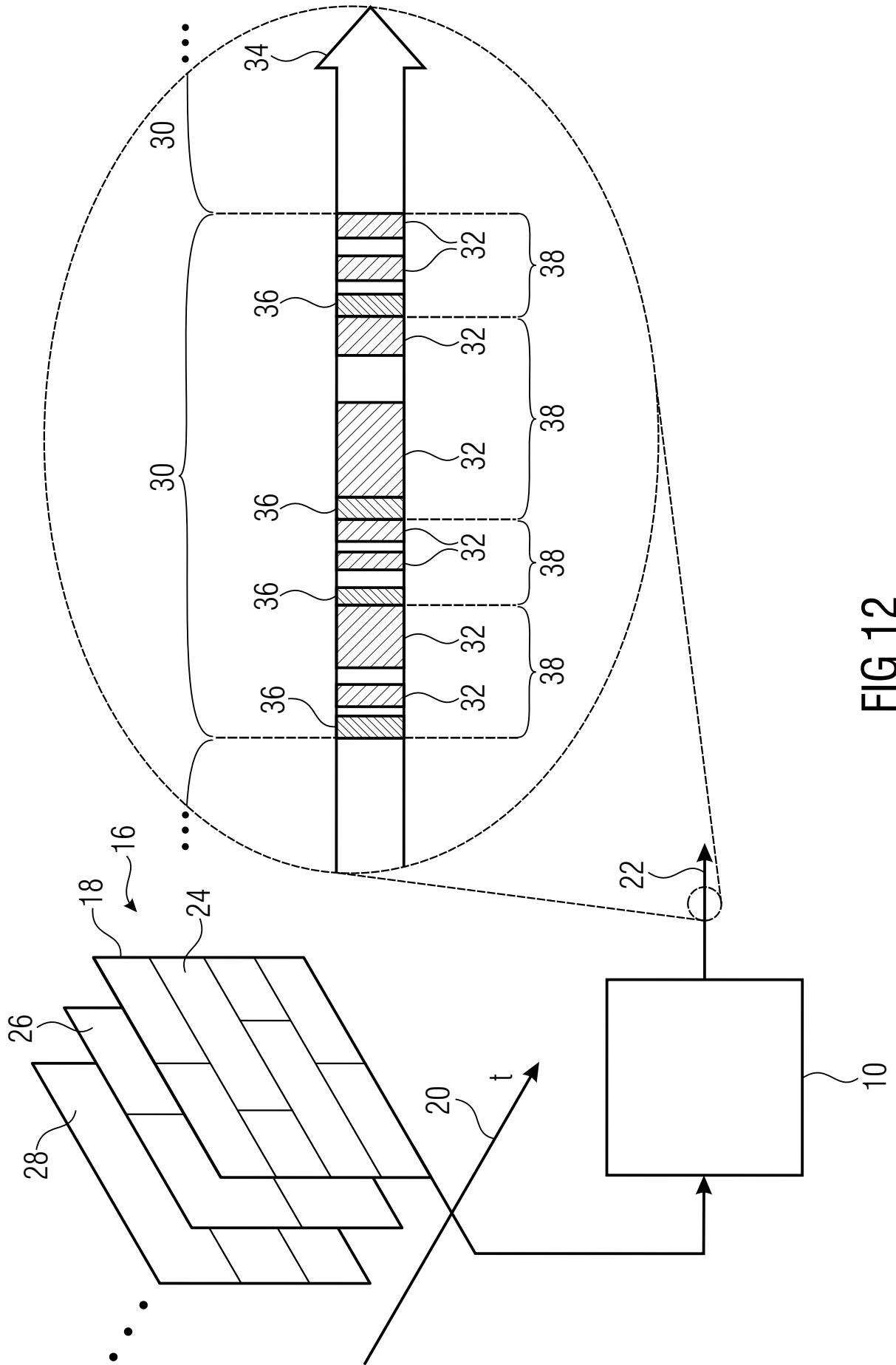


FIG 11



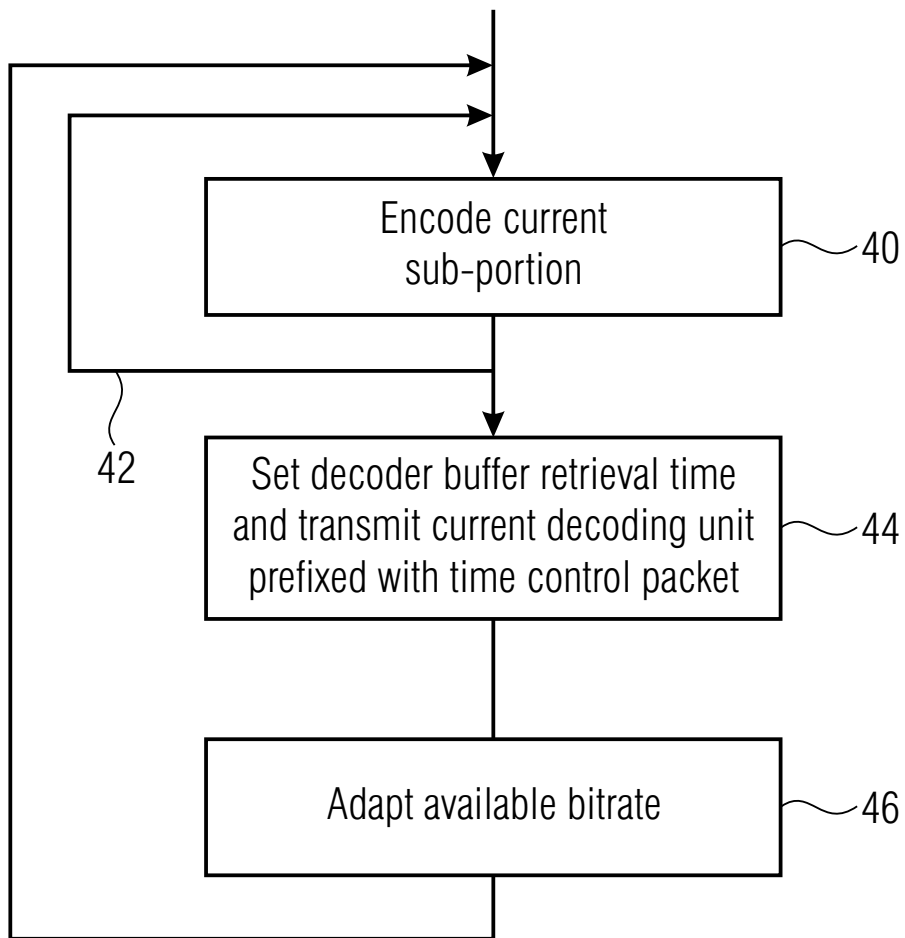


FIG 13

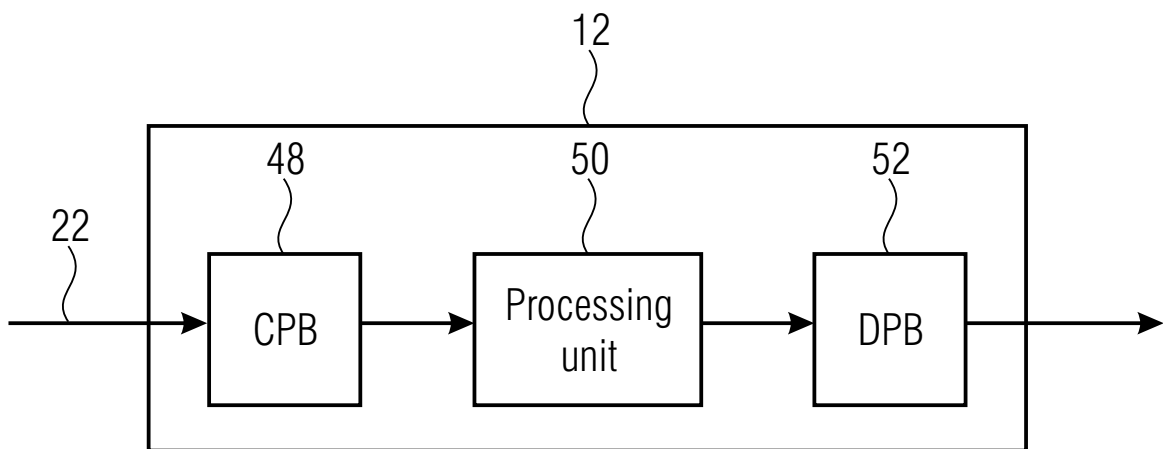
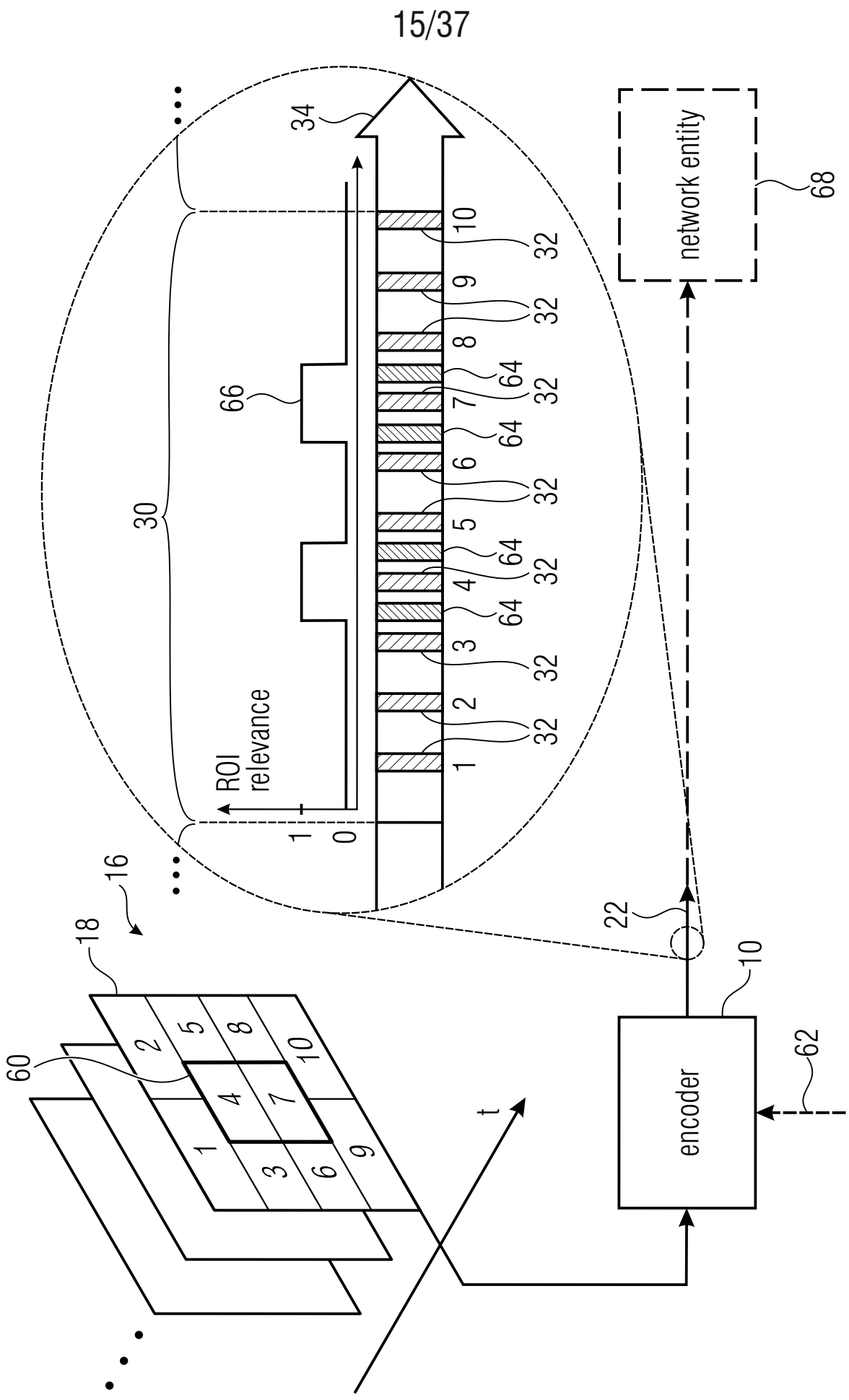
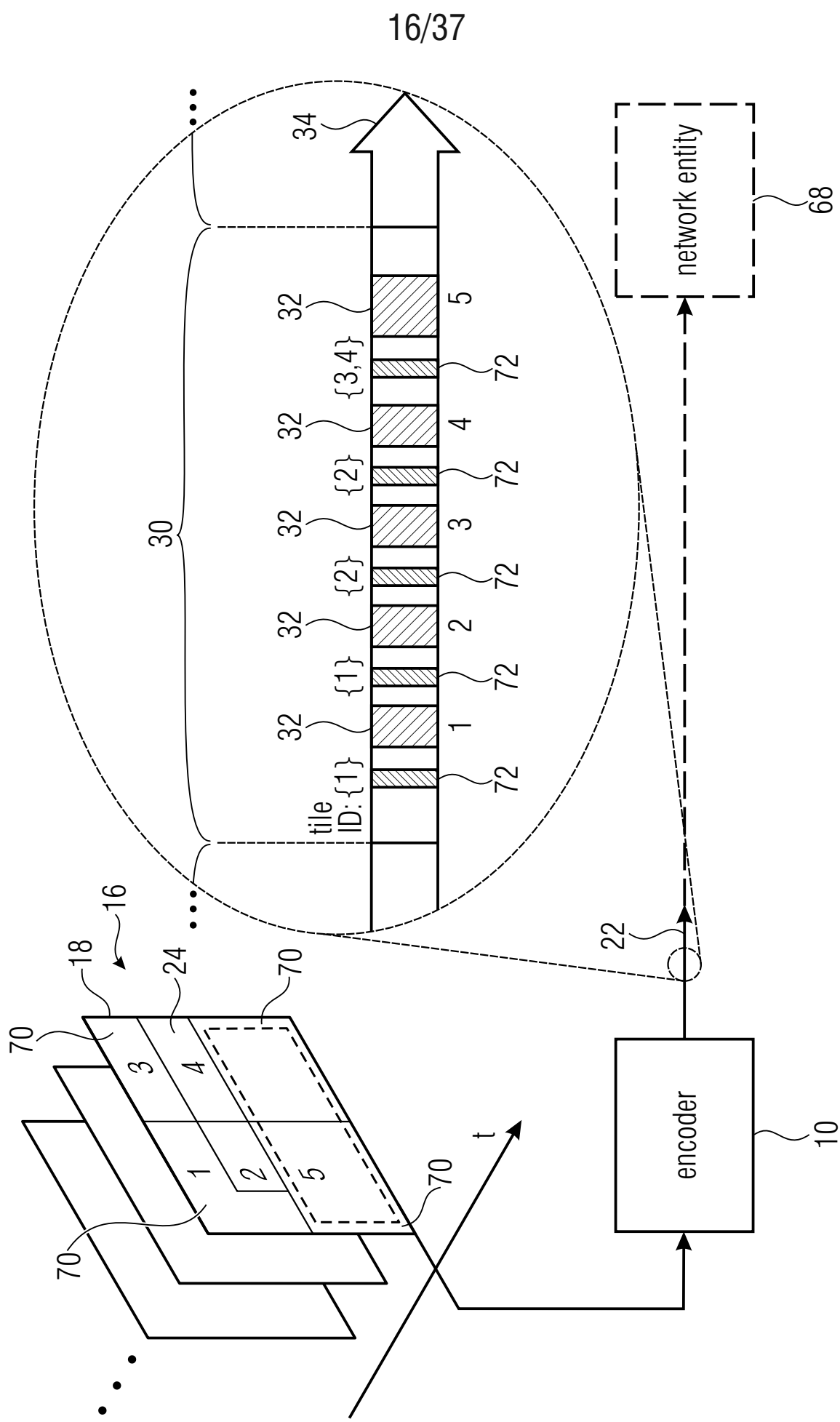


FIG 14





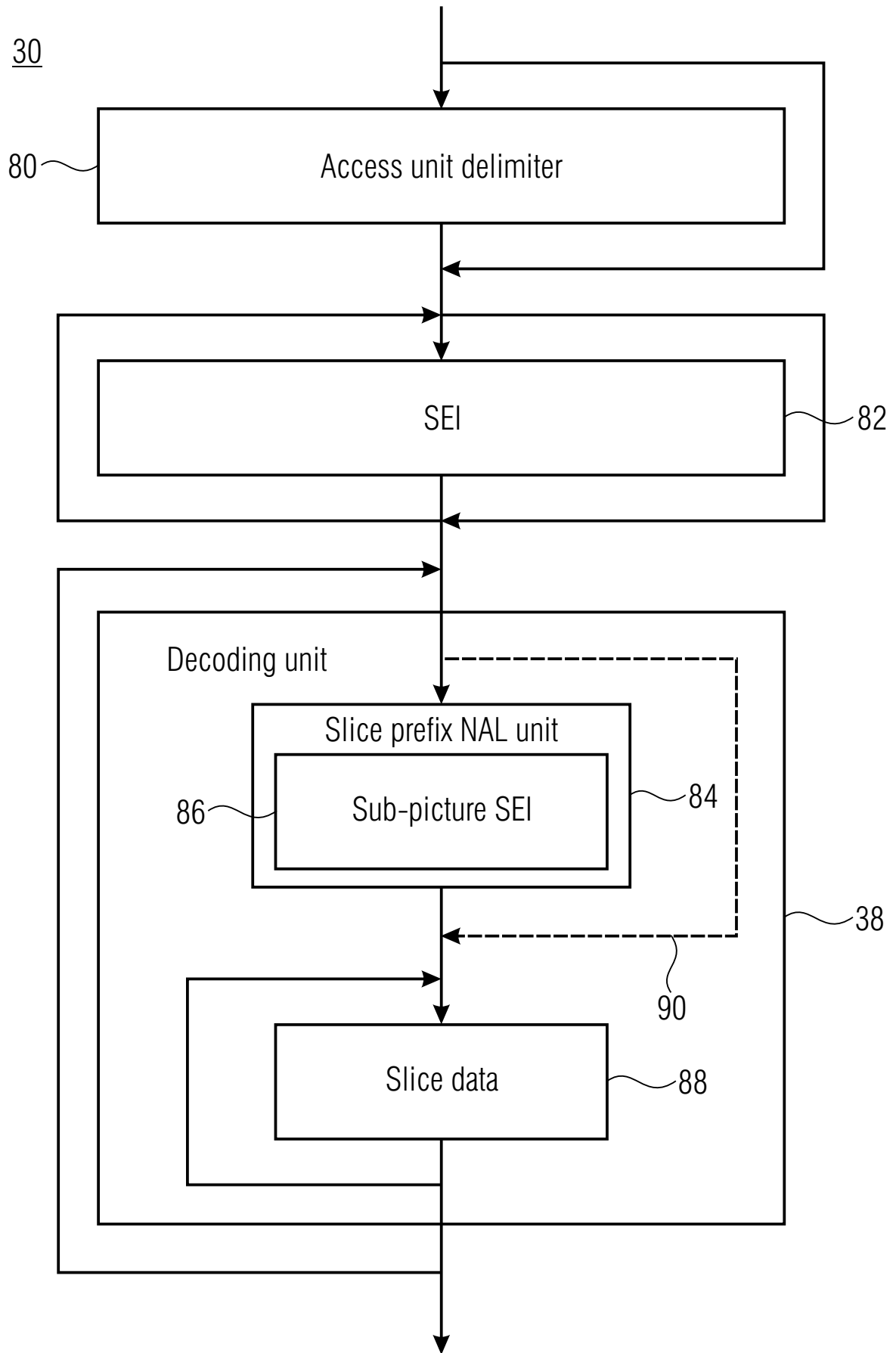


FIG 17

Tiles in region of interest

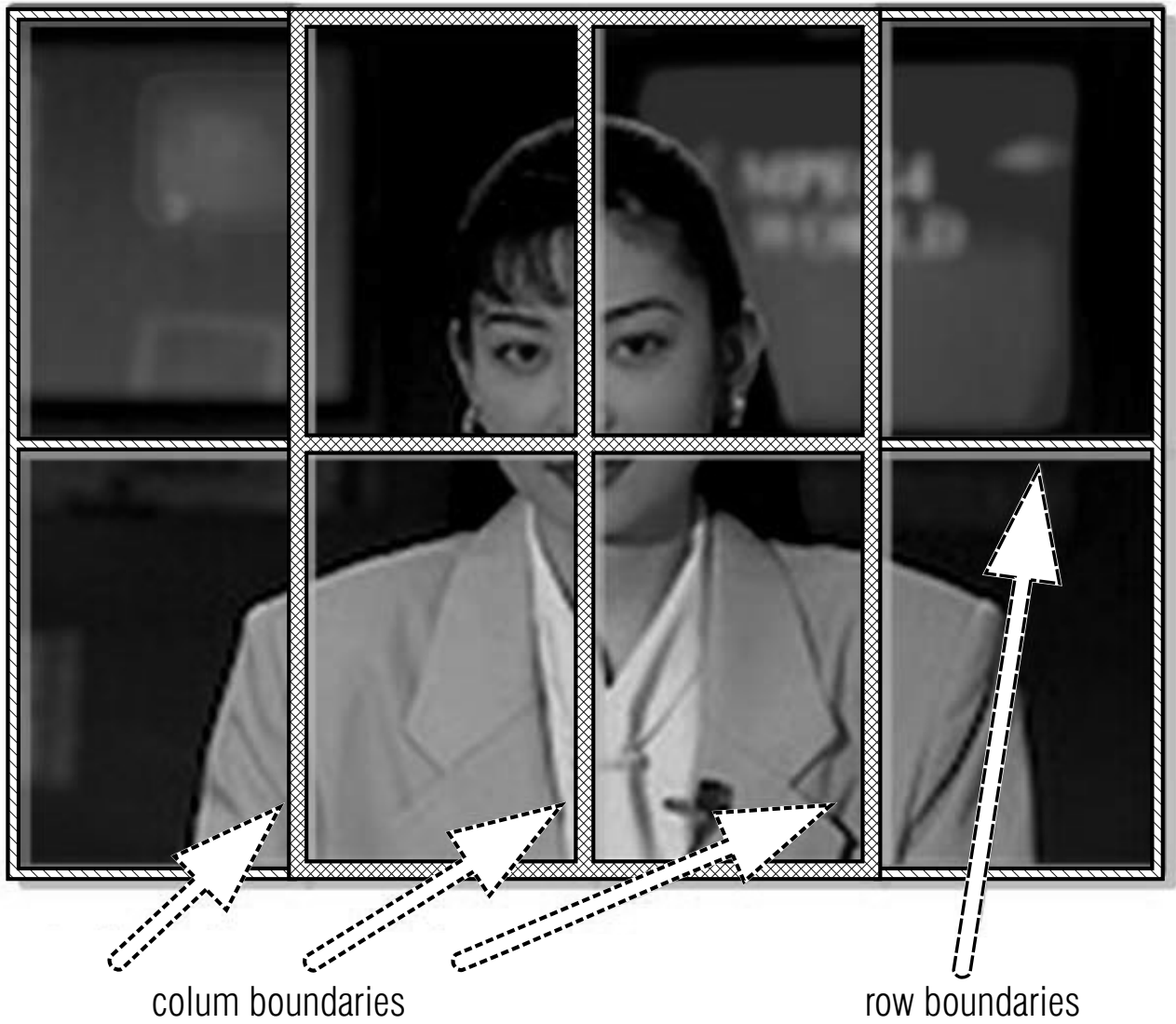


FIG 18

slice_prefix_rbsp() {	Descriptor
reserved_one_8bits	u(8)
do	
sei_message()	
while(more_rbsp_data())	
rbp_trailing_bits()	
}	

FIG 19

slice_prefix_rbsp() {	Descriptor
slice_prefix_id	u(2)
rap_flag	u(1)
decoding_unit_start_flag	u(1)
single_slice_flag	u(1)
tile_idc	u(2)
prefix_slice_header_data_present_flag	u(1)
if(prefix_slice_header_data_present_flag)	
slice_header_data()	
if(tile_idc > 0) {	
if(tile_idc == 2) /* multiple tiles in prefixed slices */	
num_tiles_in_prefixed_slices_minus1	u(16)
first_tile_id_in_prefixed_slices	u(16)
}	
do	
sei_message()	
while(more_rbsp_data())	
rbp_trailing_bits()	
}	

FIG 20

nal_unit_type	Content of NAL unit and RBSP syntax structure	NAL unit type class
0	Unspecified	non-VCL
1	Coded slice of a non-RAP, non-TFD and non-TLA picture slice_layer_rbsp()	VCL
2	Coded slice of a TFD picture slice_layer_rbsp()	VCL
3	Coded slice of a non-TFD TLA picture slice_layer_rbsp()	VCL
4, 5	Coded slice of a CRA picture slice_layer_rbsp()	VCL
6, 7	Coded slice of a BLA picture slice_layer_rbsp()	VCL
8	Coded slice of an IDR picture slice_layer_rbsp()	VCL
9..23	Reserved	n/a
24	Slice Prefix NAL unit	non-VCL
25	Video parameter set video_parameter_set_rbsp()	non-VCL
26	Sequence parameter set seq_parameter_set_rbsp()	non-VCL
27	Picture parameter set pic_parameter_set_rbsp()	non-VCL
28	Adaptation parameter set aps_rbsp()	non-VCL
29	Access unit delimiter access_unit_delimiter_rbsp()	non-VCL
30	Filler data filler_data_rbsp()	non-VCL
31	Supplemental enhancement information (SEI) sei_rbsp()	non-VCL
32..47	Reserved	n/a
48..63	Unspecified	non-VCL

FIG 21

slice_header() {	Descriptor
first_slice_in_pic_flag	u(1)
—pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
slice_header_data_present_flag	u(1)
if (slice_header_data_present_flag && !dependent_slice_flag)	
slice_header_data()	
if(tiles_or_entropy_coding_sync_idc == 1 tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte	u(8)
}	
byte_alignment()	
}	

FIG 22

slice_header_data() {	Descriptor
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(RapPicFlag) {	
rap_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
if(!IdrPicFlag) {	
pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_sps_flag	u(1)
if(!short_term_ref_pic_set_sps_flag)	
short_term_ref_pic_set(num_short_term_ref_pic_sets)	
else	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_pics; i++) {	
poc_lsb_lt[i]	u(v)
delta_poc_msb_present_flag[i]	u(1)
if(delta_poc_msb_present_flag[i])	
delta_poc_msb_cycle_lt[i]	ue(v)
used_by_curr_pic_lt_flag[i]	u(1)
}	
}	
}	
if(sample_adaptive_offset_enabled_flag) {	

FIG 23A

FIG 23

FIG 23A

FIG 23B

FIG 23C

slice_sample_adaptive_offset_flag[0]	u(1)
if(slice_sample_adaptive_offset_flag[0]) {	
slice_sample_adaptive_offset_flag[1]	u(1)
slice_sample_adaptive_offset_flag[2]	u(1)
}	
}	
if(adaptive_loop_filter_enabled_flag)	
aps_id	ue(v)
if(slice_type == P slice_type == B) {	
if(sps_temporal_mvp_enable_flag)	
pic_temporal_mvp_enable_flag	u(1)
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
if(lists_modification_present_flag)	
ref_pic_list_modification()	
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag && slice_type != I)	
cabac_init_flag	u(1)
slice_qp_delta	se(v)
if(deblocking_filter_control_present_flag) {	
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)

FIG 23B

FIG 23	FIG 23A
	FIG 23B
	FIG 23C

if(deblocking_filter_override_flag) {	
slice_header_disable_deblocking_filter_flag	u(1)
if(!slice_header_disable_deblocking_filter_flag) {	
beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	
}	
if(pic_temporal_mvp_enable_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if(slice_type != I &&	
collocated_ref_idx	ue(v)
}	
if((weighted_pred_flag && slice_type == P)	
pred_weight_table()	
if(slice_type == P slice_type == B)	
five_minus_max_num_merge_cand	ue(v)
if(adaptive_loop_filter_enabled_flag) {	
slice_adaptive_loop_filter_flag	u(1)
if(slice_adaptive_loop_filter_flag &&	
alf_param()	
if(slice_adaptive_loop_filter_flag	
alf_cu_control_param()	
}	
if(seq_loop_filter_across_slices_enabled_flag &&	
slice_loop_filter_across_slices_enabled_flag	u(1)
}	

FIG 23C

FIG 23	FIG 23A
	FIG 23B
	FIG 23C

sei_message() {	Descriptor
payloadType = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
sei_payload(payloadType, payloadSize)	
}	

FIG 24

sei_payload(payloadType, payloadSize) {	Descriptor
if(payloadType == 0)	
buffering_period(payloadSize)	
else if(payloadType == 1)	
pic_timing(payloadSize)	
else if(payloadType == 2)	
pan_scan_rect(payloadSize)	
else if(payloadType == 3)	
filler_payload(payloadSize)	
else if(payloadType == 4)	
user_data_registered_itu_t_t35(payloadSize)	
else if(payloadType == 5)	
user_data_unregistered(payloadSize)	
else if(payloadType == 6)	
recovery_point(payloadSize)	
else if(payloadType == 9)	
scene_info(payloadSize)	
else if(payloadType == 15)	
full_frame_snapshot(payloadSize)	
else if(payloadType == 16)	
progressive_refinement_segment_start(payloadSize)	
else if(payloadType == 17)	
progressive_refinement_segment_end(payloadSize)	
else if(payloadType == 19)	
film_grain_characteristics(payloadSize)	
else if(payloadType == 20)	
deblocking_filter_display_preference(payloadSize)	
else if(payloadType == 22)	
post_filter_hint(payloadSize)	
else if(payloadType == 23)	
tone_mapping_info(payloadSize)	

FIG 25A

FIG 25

FIG 25A

FIG 25B

else if(payloadType == 45)	
frame_packing_arrangement(payloadSize)	
else if(payloadType == 47)	
display_orientation(payloadSize)	
else if(payloadType == 128)	
sop_description(payloadSize)	
else if(payloadType == 129)	
field_indication(payloadSize)	
else if(payloadType == 130)	
decoded_picture_hash(payloadSize)	
else if(payloadType == 140)	
roi_info(payloadSize)	
else if(payloadType == 180)	
subpic_buffering (payloadSize)	
else if(payloadType == 181)	
subpic_timing(payloadSize)	
else if(payloadType == 182)	
subpic_tile_info (payloadSize)	
else if(payloadType == 183)	
subpic_slice_info(payloadSize)	
else if(payloadType == 184)	
subpic_tile_dimensions_info(payloadSize)	
if(!byte_aligned()) {	
bit_equal_to_one /* equal to 1 */	
while(!byte_aligned())	
bit_equal_to_zero /* equal to 0 */	
}	
}	

FIG 25B

FIG 25

FIG 25A
FIG 25B

subpic_buffering(payloadSize) {	Descriptor
seq_parameter_set_id	ue(v)
if(NalHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	

FIG 26

subpic_timing(payloadSize) {	Descriptor
reserved_one_8bits	u(8)
du_cpb_removal_delay	u(v)
du_dpb_output_delay	u(v)
}	
}	

FIG 27

subpic_slice_info(payloadSize) {	Descriptor
reserved_one_7bits	u(7)
slice_header_data_flag	u(1)
if(slice_header_data_flag)	
slice_header_data()	
byte_alignment()	
}	

FIG 28

subpic_tile_info(payloadSize) {	Descriptor
reserved_one_4bits	u(4)
tile_priority	u(3)
multiple_tiles_in_prefixed_slices_flag	u(1)
if(multiple_tiles_in_prefixed_slices_flag)	
num_tiles_in_prefixed_slices_minus1	u(16)
first_tile_id_in_prefixed_slices	u(16)
}	

FIG 29

subpic_tile_dimensions_info(payloadSize) {	Descriptor
reserved_one_7bits	u(7)
multiple_tiles_in_prefixed_slices_flag	u(1)
if(multiple_tiles_in_prefixed_slices_flag)	
num_tiles_in_prefixed_slices_minus1	u(16)
for(i = 0; i <= num_tiles_in_roi_minus1 ; i++) {	
tile_horz_start[i]	u(16)
tile_width[i]	u(16)
tile_vert_start[i]	u(16)
tile_height[i]	u(16)
}	
}	

FIG 30

roi_info(payloadSize) {	Descriptor
reserved_one_8bits	u(8)
roi_id	u(5)
roi_priority	u(3)
num_tiles_in_roi_minus1	u(16)
for(i = 0; i <= num_tiles_in_roi_minus1 ; i++) {	
roi_tile_id[i]	u(16)
}	
}	

FIG 31

roi_info(payloadSize) {	Descriptor
reserved_one_3bits	u(3)
num_rois_minus1	u(5)
for(i = 0; i <= num_roi_minus1 ; i++) {	u(5)
roi_id[i]	u(5)
roi_priority[i]	u(3)
num_tiles_in_roi_minus1[i]	u(16)
for(n = 0; n <= num_tiles_in_roi_minus1 ; n++) {	
roi_tile_id[i] [n]	u(16)
}	
roi_presentation_on_seperate_screen [i]	u(1)
}	
}	

FIG 32

decoding_unit_info(payloadSize) {	Descriptor
decoding_unit_idx	ue(v)
if(!sub_pic_cpb_params_in_pic_timing_sei_flag)	
du_spt_cpb_removal_delay_increment	u(v)
dpb_output_du_delay_present_flag	u(1)
if(dpb_output_du_delay_present_flag)	
pic_spt_dpb_output_du_delay	u(v)
}	

FIG 33

region_refresh_info(payloadSize) {	Descriptor
refreshed_region_flag	u(1)
}	

FIG 34

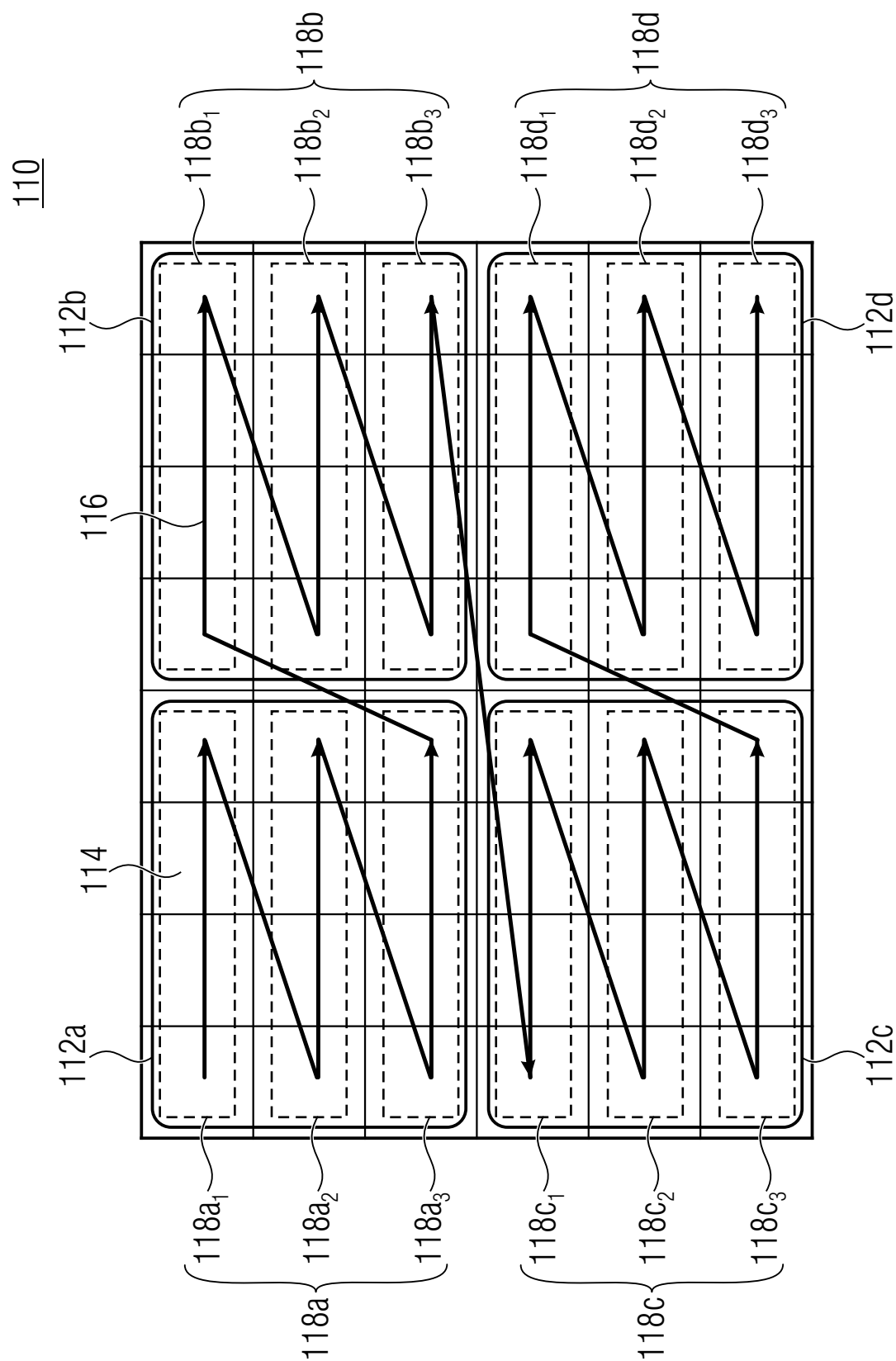


FIG 35

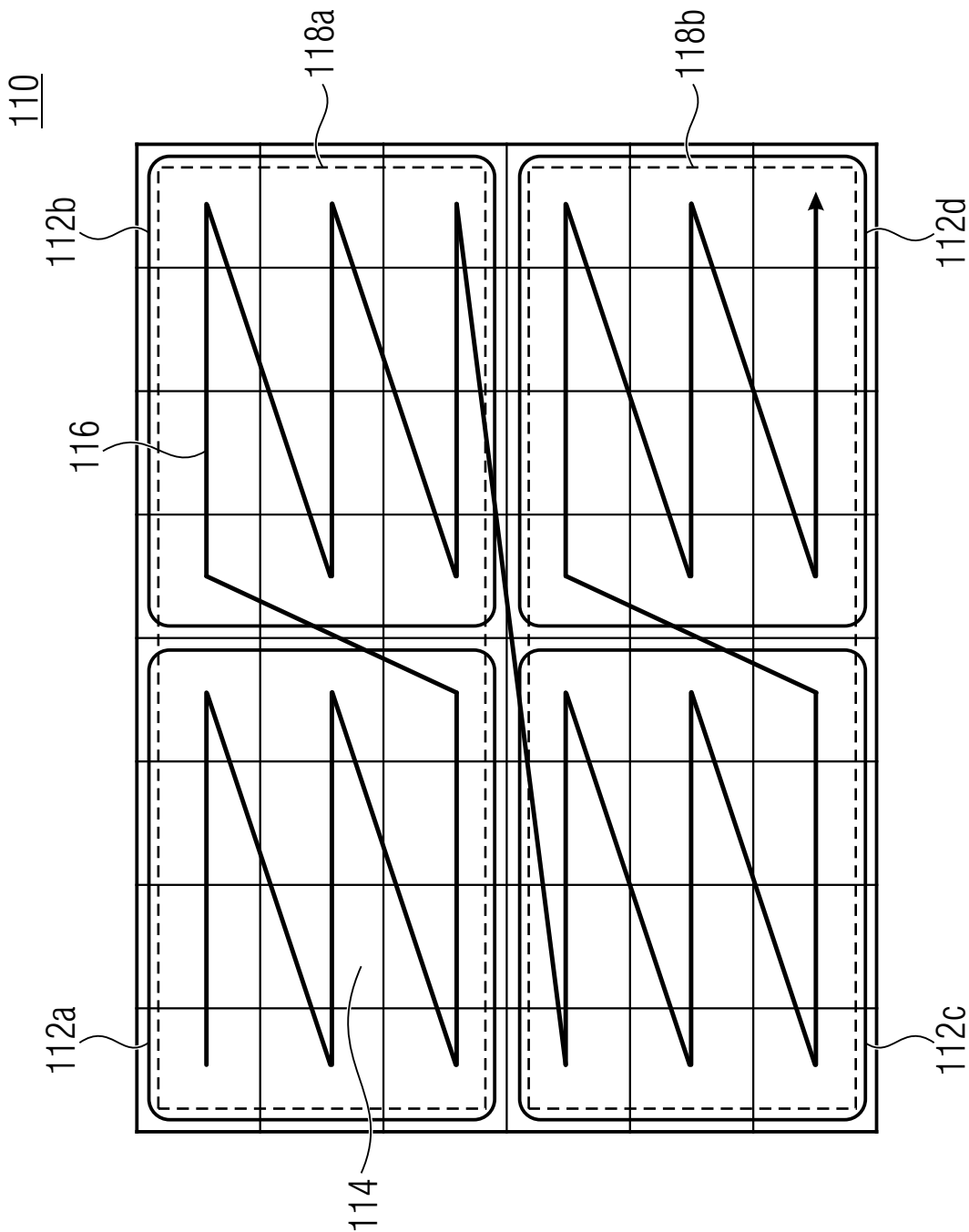


FIG 36

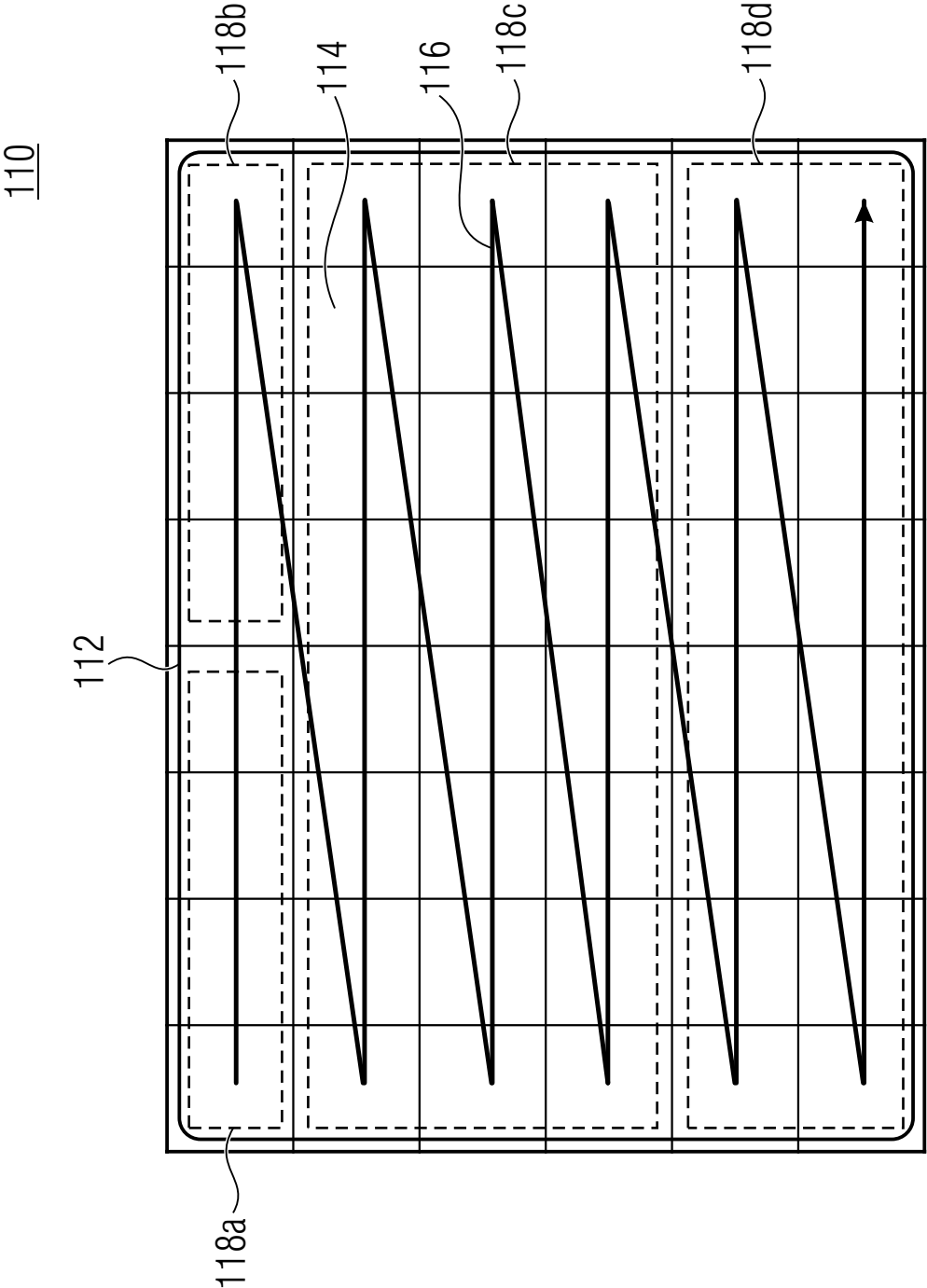


FIG 37

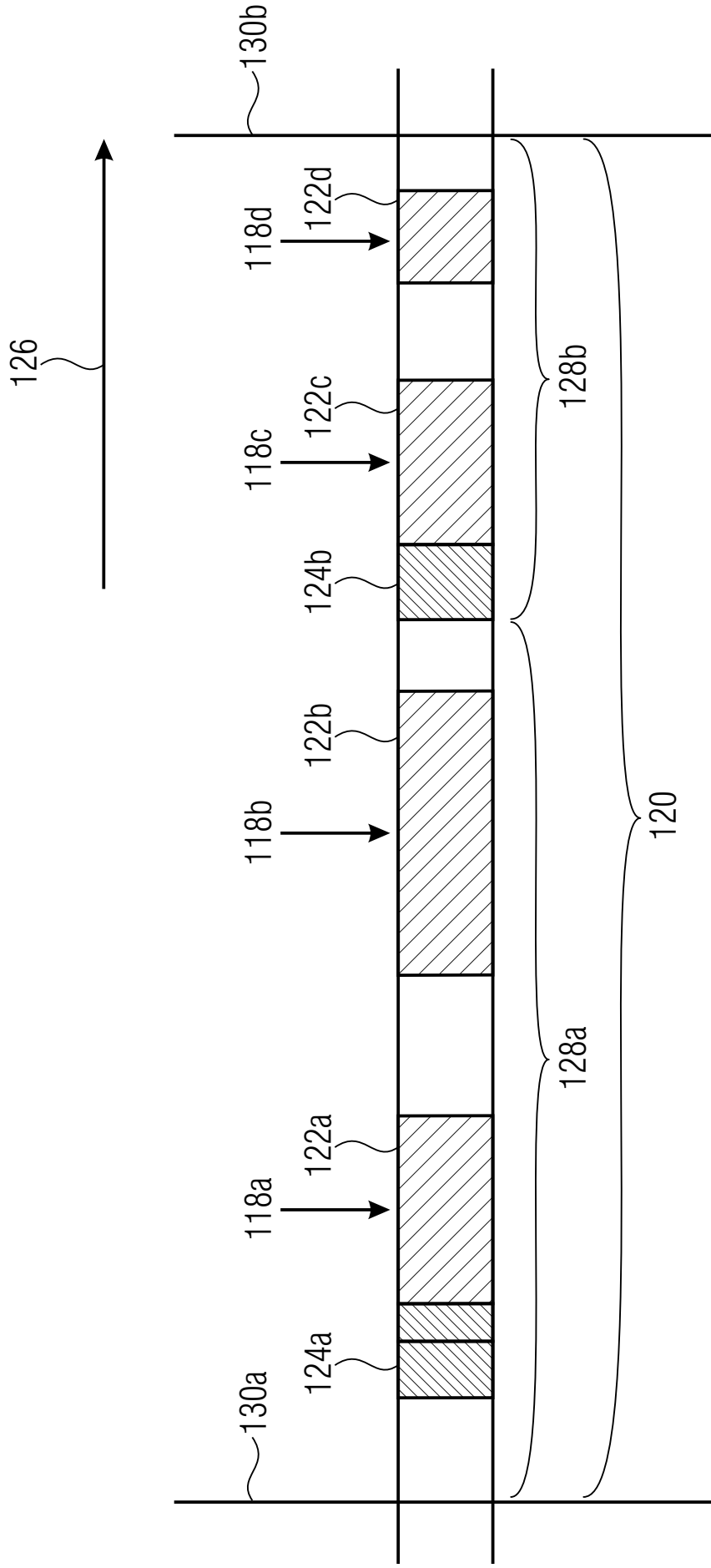


FIG 39