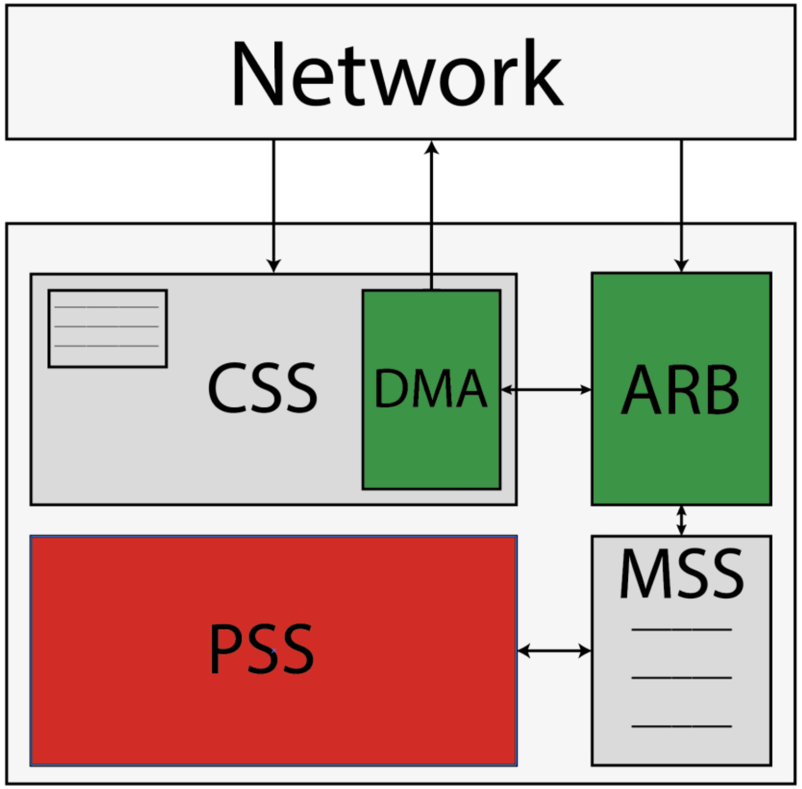
LDPC 5G decoder implementation for EMBB project

Author: Riccardo Torres

Supervisor PoliTo: Maurizio Martina

Supervisor EURECOM: Renaud Pacalet

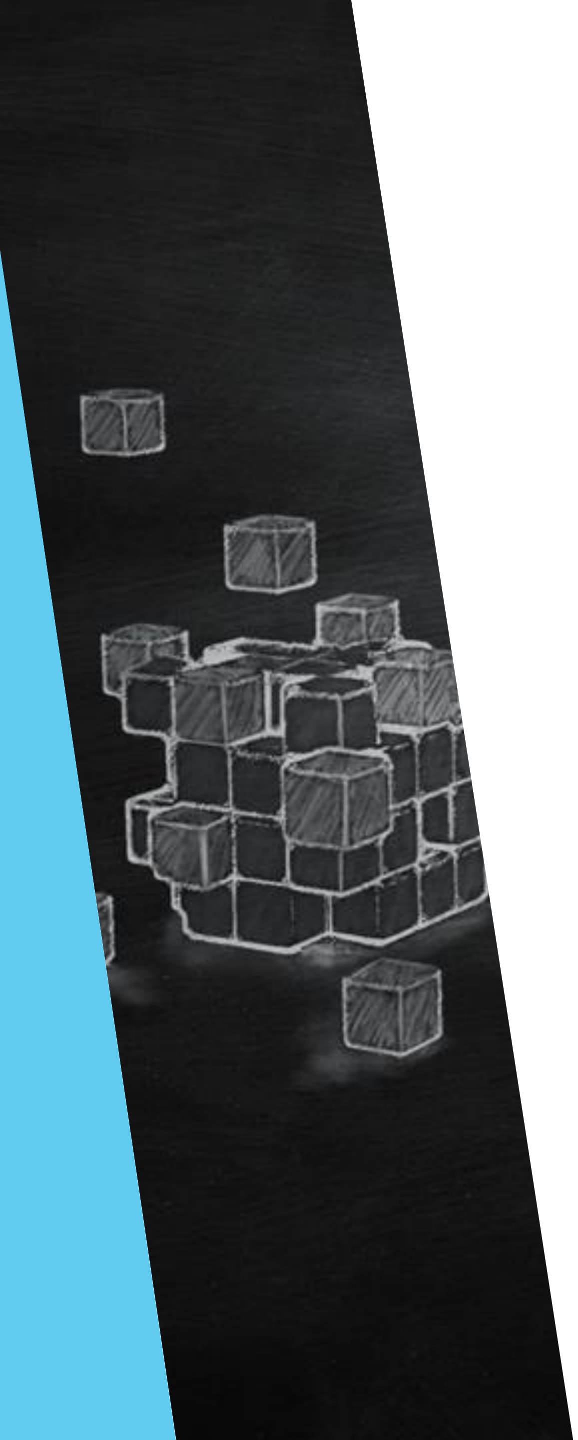


Overview of the library to be implemented

The grey blocks will not be implemented in this Project.

The two green blocks are used for handling the transfer of data inside and outside the library.

The red block is used for processing the decoding process.

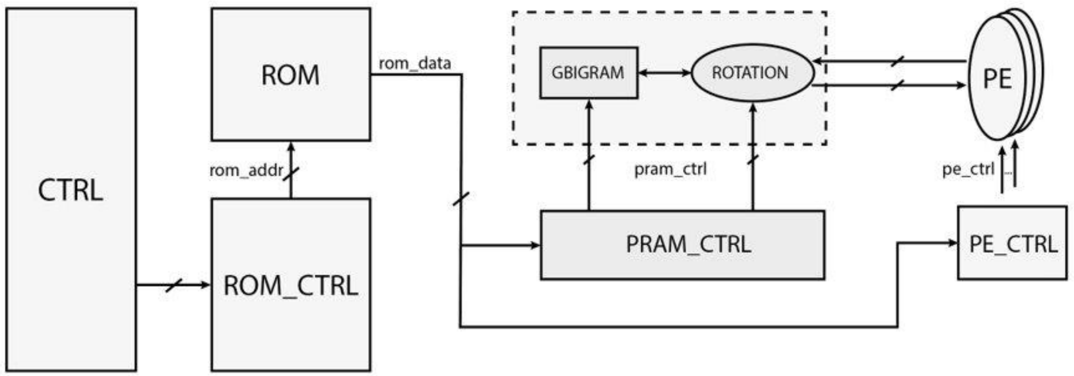


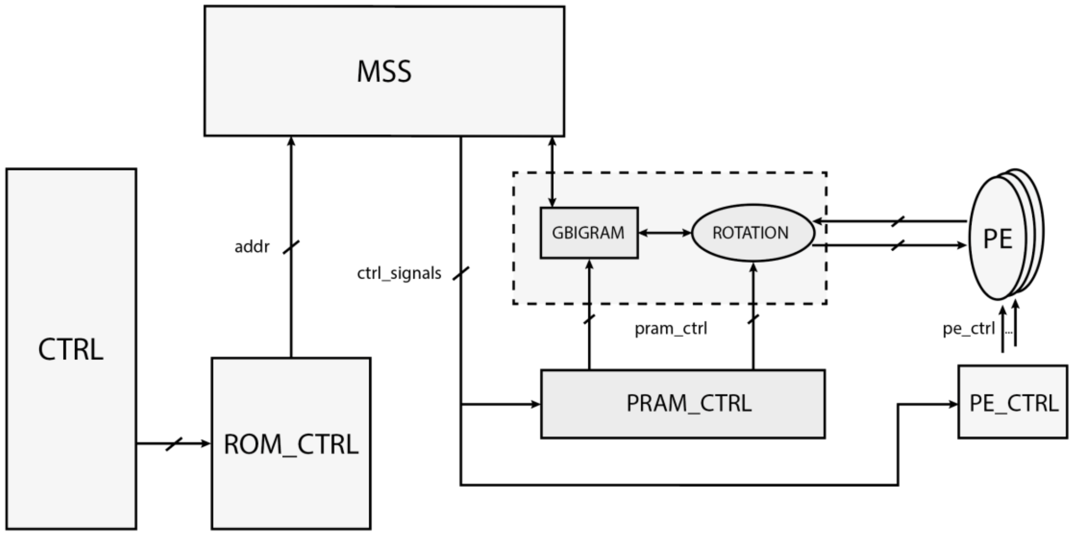
INTRODUCTION TO THE TOPICS COVERED

- The discussion of this block will be divided into the following parts:
  - **Overview** of general block structure
  - **Analysis** of the most important blocks
  - **Generation** of control signals
  - **Timing diagram** using the processing of the first two lines as an example.

Process Sub System (PSS)

- This block deals with decoding, and its structure is divided into three blocks:
  - **ROM**: where the PSS control signals are stored. The values stored in this block are actually stored in the MSS.
  - **PRAM**: where the samples to be decoded are stored.
  - **PE**: where the decoding algorithm is processed.





PROCESSING ELEMENT (PE)

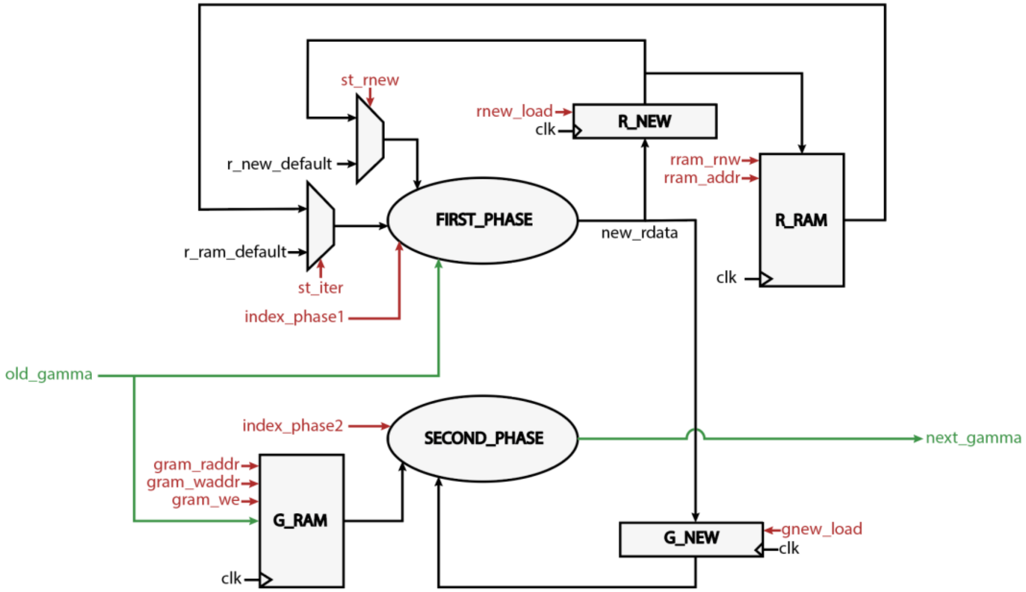
The structure of this block is intended to implement algorithm 5 shown in the figure.

The algorithm in question is the result of various optimizations; it divides the decoding work into two phases:

- The first phase runs from line 7 to line 25; it takes care of updating vectors' value, previously initialized in line 9.
- During the second phase, the new corrected values of the samples (*next_gamma*) will be generated. This correction is made using the array of values previously updated during the first phase.

As the LDPC algorithm is *iterative*, it is necessary to save the array values calculated in the first step to be reused for subsequent iterations.

The R_RAM memory is used to *store the arrays of values calculated during the first phase*, then read during subsequent iterations.



Algorithm 5 The memory-optimized SCMS algorithm with layered schedule

1: **for** $n = 0$ to $N-1$ **do** ▷ For all columns
2: $\tilde{\gamma}_n^{(0)} \leftarrow \ln(\Pr(x_n = 0|channel)/\Pr(x_n = 1|channel))$ ▷ LLRs
3: **end for**
4: **for** $m = 0$ to $M-1$ **do** ▷ For all rows
5: $r_m^{(0)} = \{0, 0, 0, \mathbf{+1}, +1, \boldsymbol{true}\}$
6: **end for**
7: **for** $l = 1$ to l_{max} **do**
8: **for** $m = 0$ to $M-1$ **do**
9: $r_m^{(l)} \leftarrow \{+\infty, +\infty, 0, \mathbf{+1}, +1, \boldsymbol{false}\}$
10: **for** $n \in \mathcal{H}(m)$ **do**
11: $\alpha \leftarrow \tilde{\gamma}_n^{(l-1)} - \beta(r_m^{(l-1)}, n)$
12: **if** $\text{sgn}(\alpha) \neq r_m^{(l-1)}.\alpha_n^s$ and (not $r_m^{(l-1)}.\alpha_n^e$) **then**
13: $\alpha \leftarrow 0$
14: $r_m^{(l)}.\alpha_n^e \leftarrow true$
15: **end if**
16: $r_m^{(l)}.\alpha_n^s \leftarrow \text{sgn}(\alpha)$
17: $r_m^{(l)}.sp \leftarrow r_m^{(l)}.sp \times \text{sgn}(\alpha)$
18: **if** $|\alpha| < r_m^{(l)}.m_1$ **then**
19: $r_m^{(l)}.m_2 \leftarrow r_m^{(l)}.m_1$
20: $r_m^{(l)}.m_1 \leftarrow |\alpha|$
21: $r_m^{(l)}.i \leftarrow n$
22: **else if** $|\alpha| < r_m^{(l)}.m_2$ **then**
23: $r_m^{(l)}.m_2 \leftarrow |\alpha|$
24: **end if**
25: **end for**
26: **for** $n \in \mathcal{H}(m)$ **do**
27: $\tilde{\gamma}_n^{(l)} \leftarrow \tilde{\gamma}_n^{(l-1)} + \beta(r_m^{(l)}, n)$
28: **end for**
29: **end for**
30: **end for**

Algorithm 4 Computing $\beta_{m,n}^{(l)}$ from $r_m^{(l)}$ compact data structure

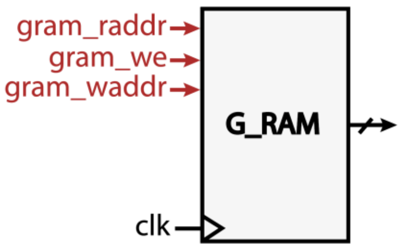
if $n = r_m^{(l)}.i$ **then**
$\beta_{m,n}^{(l)} \leftarrow r_m^{(l)}.\alpha_n^s \times r_m^{(l)}.sp \times r_m^{(l)}.m_2$
else
$\beta_{m,n}^{(l)} \leftarrow r_m^{(l)}.\alpha_n^s \times r_m^{(l)}.sp \times r_m^{(l)}.m_1$
end if

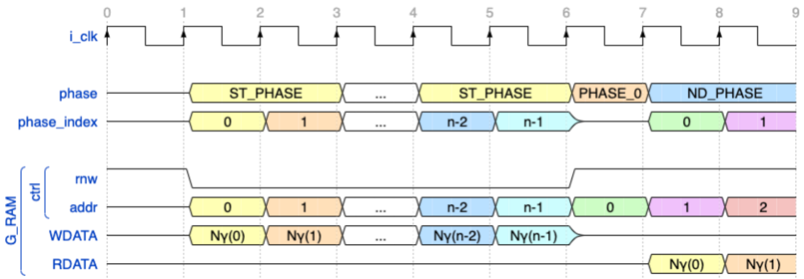
GRAM

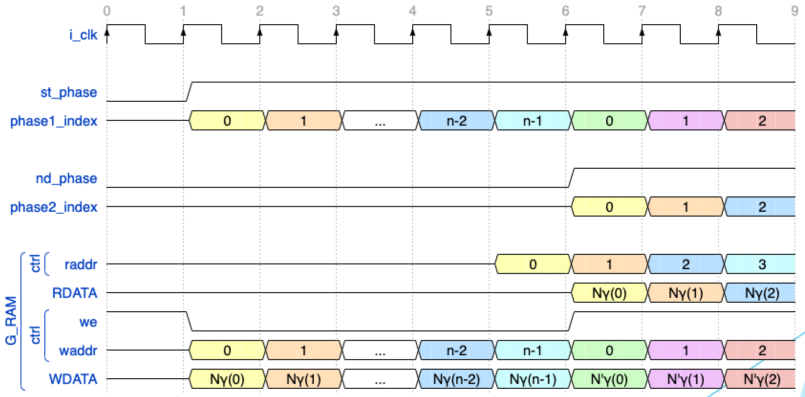
This memory is used for **storing values**, as in the second one, the previously analyzed gamma values must also be used to generate the correct values.

For this reason, a single-access memory is used within the project. Its use, however, makes it necessary to have clock strokes in which no phase is processed, and the memory is accessed in reading mode for the second phase.

To eliminate this problem and guarantee independence between the first and second phase, this memory is replaced by a **dual-port memory**.



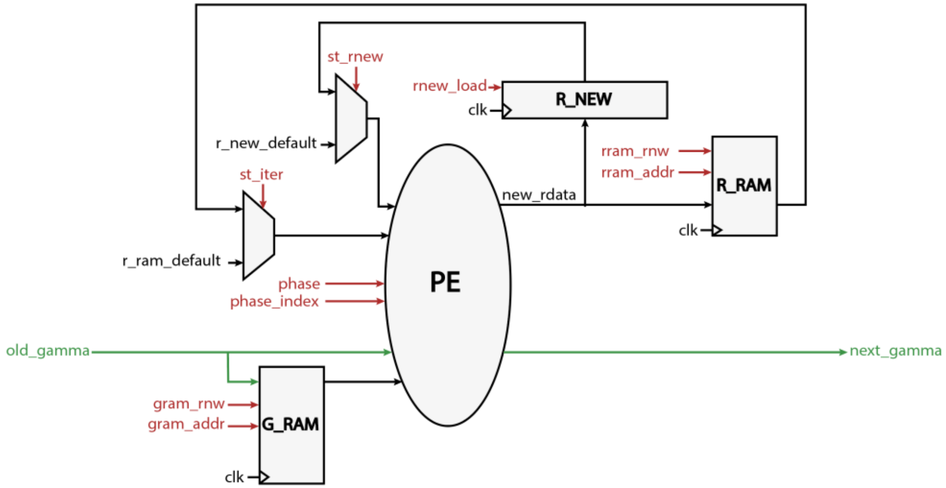


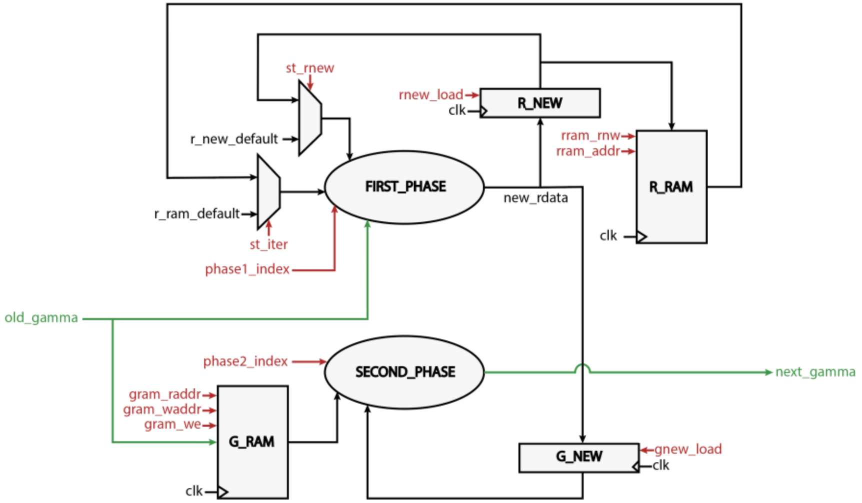


Parallelisation of the two phases

Finally, what is missing actually to *process the two phases in **parallel** is to ensure that the array of values calculated during the first phase is stored*, as needed during the second phase.

For this reason, the register **G_NEW** was introduced, which is identical to **R_NEW**, but unlike it, its value is only updated at the end of the first phase in order to store the new value array.



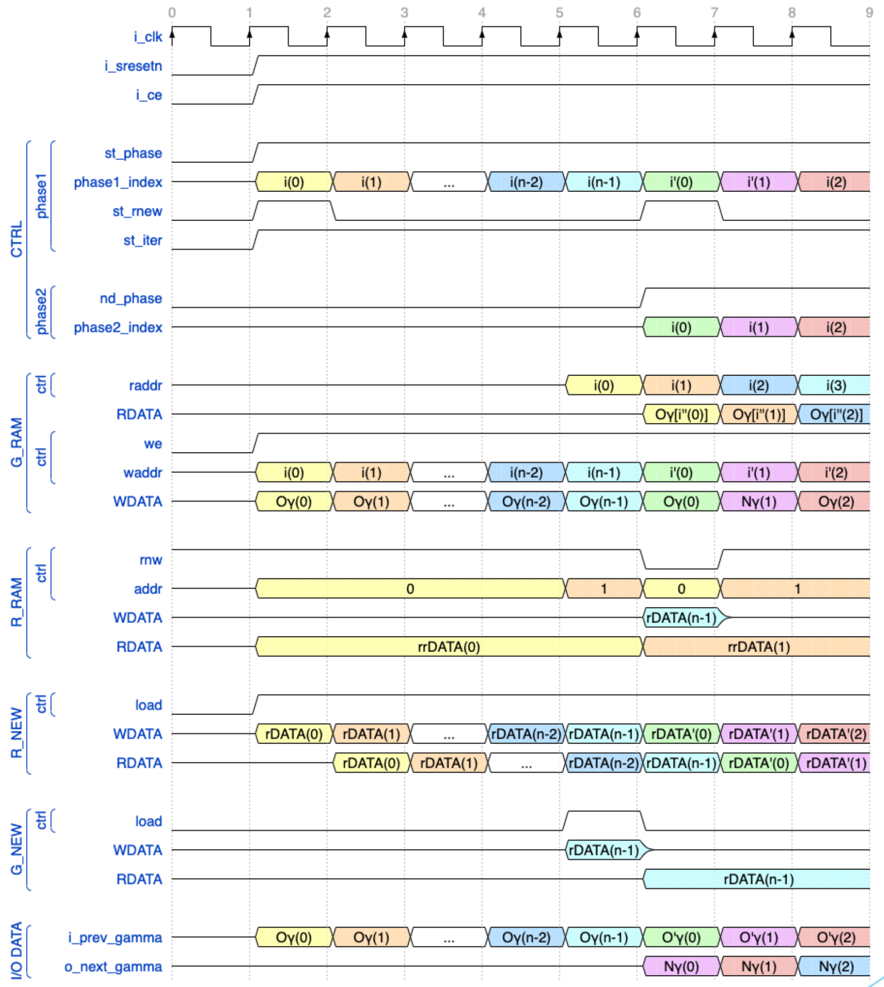


PE Timing diagram

The timing diagram opposite summarises all the features discussed above.

Following the workflow, it is interesting to note the following points:

- The constant use of both GRAM read and write operations.
- The alternation of RRAM read and write operations.
- The updating of the G_NEW_REG values at the end of the first phase.

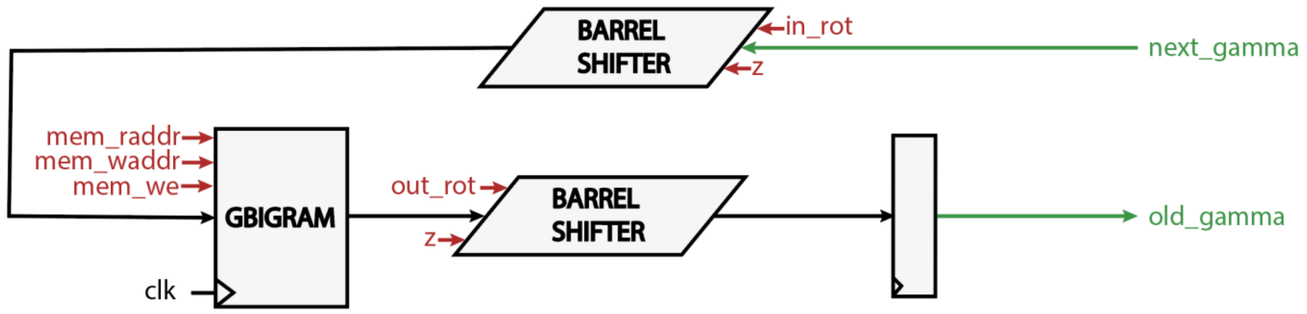


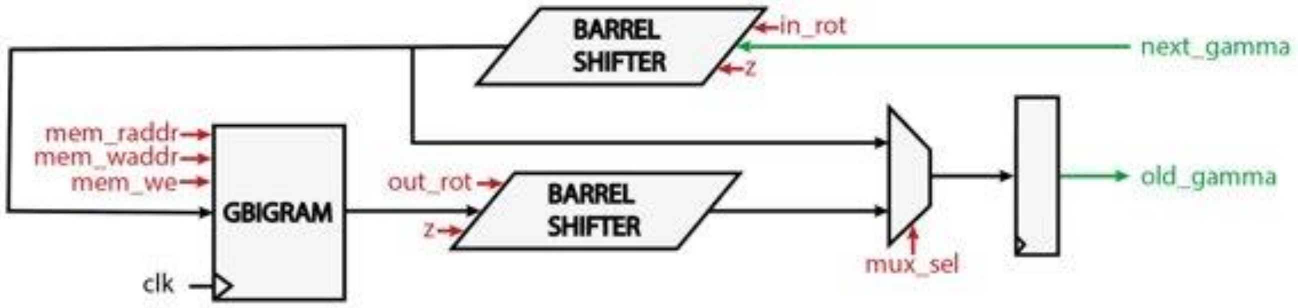
PRAM

This block handles the management of new and old gamma values sent and received, respectively from the PE.

Its initial structure, shown in the image above, is already equipped with two barrel shifters for the rotation of the values to be analyzed. This is because, to parallel the first and second phases, it will also be necessary to manage in parallel the data to be sent and those to be rewritten in GBIGRAM.

In this case, the only optimization is implementing a bypass process that allows the data coming from a previous PE processing to be analyzed directly without the need to rewrite them in memory.





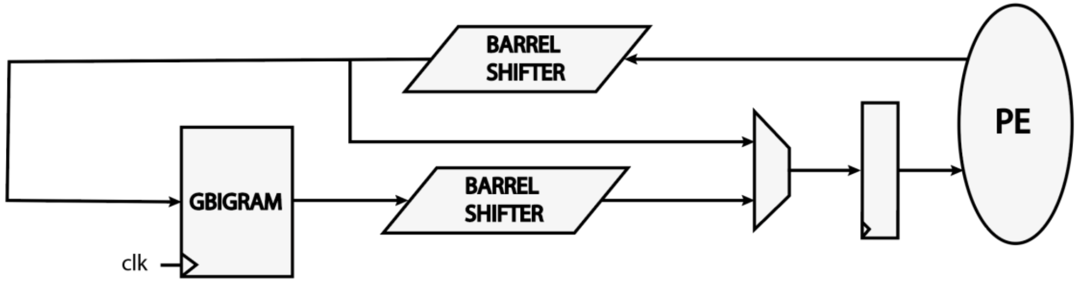
Control signals generator

The last of the three blocks is the ROM and what is interesting is not the hardware implementation but the software implementation that leads to the generation of the signals that will then be stored inside the ROM.

For this purpose, **Python code** was written to describe the behavior of the entire datapath and simulate data movement within it.

For this purpose, values can be set at program launch to set the memory elements within the circuit, leading to a delay in propagating the data.

In this case, the two memory elements register at the input of the PE and the GBIGRAM.



LDPC Base graph and its parity check matrices

index i	index j	Set index i_{LS} 0	1	2	3	4	5	6	7
0	0	250	307	73	223	211	294	0	135
	1	69	19	15	16	198	118	0	227
	2	226	50	103	94	188	167	0	126
	3	159	369	49	91	186	330	0	134
	5	100	181	240	74	219	207	0	84
	6	10	216	39	10	4	165	0	83
	9	59	317	15	0	29	243	0	53
	10	229	288	162	205	144	250	0	225
	11	110	109	215	216	116	1	0	205
	12	191	17	164	21	216	339	0	128
	13	9	357	133	215	115	201	0	75
	15	195	215	298	14	233	53	0	135
	16	23	106	110	70	144	347	0	217
	18	190	242	113	141	95	304	0	220
	19	35	180	16	198	216	167	0	90
	20	239	330	189	104	73	47	0	105
	21	31	346	32	81	261	188	0	137
	22	1	1	1	1	1	1	0	1
	23	0	0	0	0	0	0	0	0
1	0	2	76	303	141	179	77	22	96
	2	239	76	294	45	162	225	11	236
	3	117	73	27	151	223	96	124	136
	4	124	288	261	46	256	338	0	221
	5	71	144	161	119	160	268	10	128
	7	222	331	133	157	76	112	0	92
	8	104	331	4	133	202	302	0	172
	9	173	178	80	87	117	50	2	56
	11	220	295	129	206	109	167	16	11
	12	102	342	300	93	15	253	60	189
	14	109	217	76	79	72	334	0	95
	15	132	99	266	9	152	242	6	85
	16	142	354	72	118	158	257	30	153
	17	155	114	83	194	147	133	0	87
	19	255	331	260	31	156	9	168	163
	21	28	112	301	187	119	302	31	216
	22	0	0	0	0	0	0	105	0
	23	0	0	0	0	0	0	0	0
	24	0	0	0	0	0	0	0	0
2	0	106	205	68	207	258	226	132	189
	1	111	250	7	203	167	35	37	4
	2	185	328	80	31	220	213	21	225
	4	63	332	280	176	133	302	180	151
	5	117	256	38	180	243	111	4	236
	6	93	161	227	186	202	265	149	117
	7	229	267	202	95	218	128	48	179
	8	177	160	200	153	63	237	38	92
	9	95	63	71	177	0	294	122	24
	10	39	129	106	70	3	127	195	68
	13	142	200	295	77	74	110	155	6
	14	225	88	283	214	229	286	28	101
	15	225	53	301	77	0	125	85	33
	17	245	131	184	198	216	131	47	96
	18	205	240	246	117	269	163	179	125
	19	251	205	230	223	200	210	42	67
	20	117	13	276	90	234	7	66	230
	24	0	0	0	0	0	0	0	0
	25	0	0	0	0	0	0	0	0
3	0	121	276	220	201	187	97	4	128
	1	89	87	208	18	145	94	6	23
	3	84	0	30	165	166	49	33	162
	4	20	275	197	5	108	279	113	220
	6	150	199	61	45	82	139	49	43
	7	131	153	175	142	132	166	21	186
	8	243	56	79	16	197	91	6	96
	10	136	132	281	34	41	106	151	1
	11	86	305	303	155	162	246	83	216
	12	246	231	253	213	57	345	154	22
	13	219	341	164	147	36	269	87	24
	14	211	212	53	69	115	185	5	167
	16	240	304	44	96	242	249	92	200
	17	76	300	28	74	165	215	173	32
	18	244	271	77	99	0	143	120	235
	20	144	39	319	30	113	121	2	172
	21	12	357	68	158	108	121	142	219
	22	1	1	1	1	1	1	0	1
	25	0	0	0	0	0	0	0	0
4	0	157	332	233	170	246	42	24	64
	1	102	181	205	10	235	256	204	211
	26	0	0	0	0	0	0	0	0
5	0	205	195	83	164	261	219	185	2
	1	236	14	292	59	181	130	100	171
	3	194	115	50	86	72	251	24	47
	12	231	166	318	80	283	322	65	143
	16	28	241	201	182	254	295	207	210
15	1	96	2	290	120	0	348	6	138
	10	65	210	60	131	183	15	81	220
	13	63	318	130	209	108	81	182	173
	18	75	55	184	209	68	176	53	142
	25	179	269	51	81	64	113	46	49
	37	0	0	0	0	0	0	0	0
16	1	64	13	69	154	270	190	88	78
	3	49	338	140	164	13	293	198	152
	11	49	57	45	43	99	332	160	84
	20	51	289	115	189	54	331	122	5
	22	154	57	300	101	0	114	182	205
	38	0	0	0	0	0	0	0	0
17	0	7	260	257	56	153	110	91	183
	14	164	303	147	110	137	228	184	112
	16	59	81	128	200	0	247	30	106
	17	1	358	51	63	0	116	3	219
	21	144	375	228	4	162	190	155	129
	39	0	0	0	0	0	0	0	0
18	1	42	130	260	199	161	47	1	183
	12	233	163	294	110	151	286	41	215
	13	8	280	291	200	0	246	167	180
	18	155	132	141	143	241	181	68	143
	19	147	4	295	186	144	73	148	14
	40	0	0	0	0	0	0	0	0
19	0	60	145	64	8	0	87	12	179
	1	73	213	181	6	0	110	6	108
	7	72	344	101	103	118	147	166	159
	8	127	242	270	198	144	258	184	138
	10	224	197	41	8	0	204	191	196
	41	0	0	0	0	0	0	0	0
20	0	151	187	301	105	265	89	6	77
	3	186	206	162	210	81	65	12	187
	9	217	264	40	121	90	155	15	203
	11	47	341	130	214	144	244	5	167
	22	160	59	10	183	228	30	30	130
	42	0	0	0	0	0	0	0	0
21	1	249	205	79	192	64	162	6	197
	5	121	102	175	131	46	264	86	122
	16	109	328	132	220	266	346	96	215
	20	131	213	283	50	9	143	42	65
	21	171	97	103	106	18	109	199	216
	43	0	0	0	0	0	0	0	0
22	0	64	30	177	53	72	280	44	25
	12	142	11	20	0	189	157	58	47
	13	188	233	55	3	72	236	130	126
	17	158	22	316	148	257	113	131	178
	44	0	0	0	0	0	0	0	0
23	1	156	24	249	88	180	18	45	185
	2	147	89	50	203	0	6	18	127
	10	170	61	133	168	0	181	132	117
	18	152	27	105	122	165	304	100	199
	45	0	0	0	0	0	0	0	0
24	0	112	298	289	49	236	38	9	32
	3	86	158	280	157	199	170	125	178
	4	236	235	110	64	0	249	191	2
	11	116	339	187	193	266	288	28	156
	22	222	234	281	124	0	194	6	58
	46	0	0	0	0	0	0	0	0
25	1	23	72	172	1	205	279	4	27
	6	136	17	295	166	0	255	74	141
	7	116	383	96	65	0	111	16	11
	14	182	312	46	81	183	54	28	181
	47	0	0	0	0	0	0	0	0
26	0	195	71	270	107	0	325	21	163
	2	243	81	110	176	0	326	142	131
	4	215	76	318	212	0	226	192	169
	15	61	136	67	127	277	99	197	98
	48	0	0	0	0	0	0	0	0
27	1	25	194	210	208	45	91	98	165
	6	104	194	29	141	36	326	140	232
	8	194	101	304	174	72	268	22	9
	49	0	0	0	0	0	0	0	0
28	0	128	222	11	146	275	102	4	32
	4	165	19	293	153	0	1	1	43
	19	181	244	50	217	155	40	40	200
	21	63	274	234	114	62	167	93	205
	50	0	0	0	0	0	0	0	0
29	1	86	252	27	150	0	273	92	232
	14	236	5	308	11	180	104	136	32
	18	84	147	117	53	0	243	106	118
	25	6	78	29	68	42	107	6	103
	51	0	0	0	0	0	0	0	0
30	0	216	159	91	34	0	171	2	170
	10	73	229	23	130	90	16	88	199
	21	123	51	267	130	79	258	161	180
	22	115	157	279	153	144	283	72	180
	27	0	0	0	0	0	0	0	0
6	0	183	278	289	158	80	294	6	199
	6	22	257	21	119	144	73	27	22
	10	28	1	293	113	169	330	163	23
	11	67	351	13	21	90	99	50	100
	13	244	92	232	63	59	172	48	92
	17	11	253	302	51	177	150	24	207
	18	157	18	138	136	151	284	38	52
	20	211	225	235	116	108	305	91	13
	28	0	0	0	0	0	0	0	0
7	0	220	9	12	17	169	3	145	77
	1	44	62	88	76	189	103	88	146
	4	159	316	207	104	154	224	112	209
	7	31	333	50	100	184	297	153	32
	8	167	290	25	150	104	215	159	166
	14	104	114	76	158	164	39	76	18
	29	0	0	0	0	0	0	0	0
8	0	112	307	295	33	54	348	172	181
	1	4	179	133	95	0	75	2	105
	3	7	165	130	4	252	22	131	141
	12	211	18	231	217	41	312	141	223
	16	102	39	296	204	98	224	96	177
	19	164	224	110	39	46	17	99	145
	21	109	368	269	58	15	59	101	199
	22	241	67	245	44	230	314	35	153
	24	90	170	154	201	54	244	116	38
	30	0	0	0	0	0	0	0	0
9	0	103	366	189	9	162	156	6	169
	1	182	232	244	37	159	88	10	12
	10	109	321	36	213	93	293	145	206
	11	21	133	286	105	134	111	53	221
	13	142	57	151	89	45	92	201	17
	17	14	303	267	185	132	152	4	212
	18	61	63	135	109	76	23	164	92
	20	216	82	209	218	209	337	173	205
	31	0	0	0	0	0	0	0	0
10	1	98	101	14	82	178	175	126	116
	2	149	339	80	165	1	253	77	151
	4	167	274	211	174	28	27	156	70
	7	160	111	75	19	267	231	16	230
	8	49	383	161	194	234	49	12	115
	14	58	354	311	103	201	267	70	84
	32	0	0	0	0	0	0	0	0
11	0	77	48	16	52	55	25	184	45
	1	41	102	147	11	23	322	194	115
	12	83	8	290	2	274	200	123	134
	16	182	47	289	35	181	351	16	1
	21	78	188	177	32	273	166	104	152
	22	252	334	43	84	39	338	109	165
	23	22	115	280	201	26	192	124	107
	33	0	0	0	0	0	0	0	0
12	0	160	77	229	142	225	123	6	186
	1	42	186	235	175	162	217	20	215
	10	21	174	169	136	244	142	203	124
	11	32	232	48	3	151	110	153	180
	13	234	50	105	28	238	176	104	98
	18	7	74	52	182	243	76	207	80
	34	0	0	0	0	0	0	0	0
13	0	177	313	39	81	231	311	52	220
	3	248	177	302	56	0	251	147	185
	7	151	266	303	72	216	265	1	154
	20	185	115	160	217	47	94	16	178
	23	62	370	37	78	36	81	46	150
	35	0	0	0	0	0	0	0	0
14	0	206	142	78	14	0	22	1	124
	12	55	248	299	175	186	322	202	144
	15	206	137	54	211	253	277	118	182
	16	127	89	61	191	16	156	130	95
	17	16	347	179	51	0	66	1	72
	21	229	12	258	43	79	78	2	76
	36	0	0	0	0	0	0	0	0
15	0	40	241	229	90	170	176	173	39
	13	120	260	105	210	252	95	112	26
	24	9	90	135	123	173	212	20	105
	52	0	0	0	0	0	0	0	0
31	1	95	100	222	175	144	101	4	73
	7	177	215	308	49	144	297	49	149
	22	172	258	66	177	166	279	125	175
	25	61	256	162	128	19	222	194	108
	53	0	0	0	0	0	0	0	0
32	0	221	102	210	192	0	351	6	103
	12	112	201	22	209	211	265	126	110
	14	199	175	271	58	36	338	63	151
	24	121	287	217	30	162	83	20	211
	54	0	0	0	0	0	0	0	0
33	1	2	323	170	114	0	56	10	199
	2	187	8	20	49	0	304	30	132
	11	41	361	140	161	76	141	6	172
	21	211	105	33	137	18	101	92	65
	55	0	0	0	0	0	0	0	0
34	0	127	230	187	82	197	60	4	161
	7	167	148	296	186	0	320	153	237
	15	164	202	5	68	108	112	197	142
	17	159	312	44	150	0	54	155	180
	56	0	0	0	0	0	0	0	0
35	1	161	320	207	192	199	100	4	231
	6	197	335	158	173	278	210	45	174
	12	207	2	55	26	0	195	168	145
	22	103	266	285	187	205	268	185	100
	57	0	0	0	0	0	0	0	0
36	0	37	210	259	222	216	135	6	11
	14	105	313	179	157	16	15	200	207
	15	51	297	178	0	0	35	177	42
	18	120	21	160	6	0	188	43	100
	58	0	0	0	0	0	0	0	0
37	1	198	269	298	81	72	319	82	59
	13	220	82	15	195	144	236	2	204
	23	122	115	115	138	0	85	135	161
	59	0	0	0	0	0	0	0	0
38	0	167	185	151	123	190	164	91	121
	9	151	177	179	90	0	196	64	90
	10	157	289	64	73	0	209	198	26
	12	163	214	181	10	0	246	100	140
	60	0	0	0	0	0	0	0	0
39	1	173	258	102	12	153	236	4	115
	3	139	93	77	77	0	264	28	188
	7	149	346	192	49	165	37	109	168
	19	0	297	208	114	117	272	188	52
	61	0	0	0	0	0	0	0	0
40	0	157	175	32	67	216	304	10	4
	8	137	37	80	45	144	237	84	103
	17	149	312	197	96	2	135	12	30
	62	0	0	0	0	0	0	0	0
41	1	167	52	154	23	0	123	2	53
	3	173	314	47	215	0	77	75	189
	9	139	139	124	60	0	25	142	215
	18	151	288	207	167	183	272	128	24
	63	0	0	0	0	0	0	0	0
42	0	149	113	226	114	27	288	163	222
	4	157	14	65	91	0	83	10	170
	24	137	218	126	78	35	17	162	71
	64	0	0	0	0	0	0	0	0
43	1	151	113	228	206	52	210	1	22
	16	163	132	69	22	243	3	163	127
	18	173	114	176	134	0	53	99	49
	25	139	168	102	161	270	167	98	125
	65	0	0	0	0	0	0	0	0
44	0	139	80	234	84	18	79	4	191
	7	157	78	227	4	0	244	6	211
	9	163	163	259	9	0	293	142	187
	22	173	274	260	12	57	272	3	148
	66	0	0	0	0	0	0	0	0
45	1	149	135	101	184	168	82	181	177
	6	151	149	228	121	0	67	45	114
	10	167	15	126	29	144	235	153	93
	67	0	0	0	0	0	0	0	0

These parameters define which values are to be analyzed for decoding.

In particular, depending on the lifting size value chosen, it is possible to extrapolate the **rotation value** associated with each column.

In this case, the value z = 384 was chosen, which corresponds to the maximum potential rotation value.

This value falls within the second set and therefore, the values used will be those of the second column (set index = 1).

3	0	121	276	220	201	187	97	4	128
	1	89	87	208	18	145	94	6	23
	3	84	0	30	165	166	49	33	162
	4	20	275	197	5	108	279	113	220
	6	150	199	61	45	82	139	49	43
	7	131	153	175	142	132	166	21	186
	8	243	56	79	16	197	91	6	96
	10	136	132	281	34	41	106	151	1
	11	86	305	303	155	162	246	83	216
	12	246	231	253	213	57	345	154	22
	13	219	341	164	147	36	269	87	24
	14	211	212	53	69	115	185	5	167
	16	240	304	44	96	242	249	92	200
	17	76	300	28	74	165	215	173	32
	18	244	271	77	99	0	143	120	235
	20	144	39	319	30	113	121	2	172
	21	12	357	68	158	108	121	142	219
	22	1	1	1	1	1	1	0	1
	25	0	0	0	0	0	0	0	0
4	0	157	332	233	170	246	42	24	64
	1	102	181	205	10	235	256	204	211
	26	0	0	0	0	0	0	0	0

4	0	157	332	233	170	246	42	24	64
	1	102	181	205	10	235	256	204	211
	26	0	0	0	0	0	0	0	0
5	0	205	195	83	164	261	219	185	2
	1	236	14	292	59	181	130	100	171
	3	194	115	50	86	72	251	24	47
	12	231	166	318	80	283	322	65	143
	16	28	241	201	182	254	295	207	210
	21	123	51	267	130	79	258	161	180
	22	115	157	279	153	144	283	72	180
	27	0	0	0	0	0	0	0	0

It is important to note that analyzing the rows in the order described by the base graph leads to situations where, when moving from one row to another, there is a significant difference in the number of columns analyzed.

This means that it will be necessary to stop one of the two phases in many cases because it is **waiting for the other to finish.**

This problem is solved by **reordering the rows** at the beginning of the process.

0	0	250	307	73	223	211	294	0	135
	1	69	19	15	16	198	118	0	227
	2	226	50	103	94	188	167	0	126
	3	159	369	49	91	186	330	0	134
	5	100	181	240	74	219	207	0	84
	6	10	216	39	10	4	165	0	83
	9	59	317	15	0	29	243	0	53
	10	229	288	162	205	144	250	0	225
	11	110	109	215	216	116	1	0	205
	12	191	17	164	21	216	339	0	128
	13	9	357	133	215	115	201	0	75
	15	195	215	298	14	233	53	0	135
	16	23	106	110	70	144	347	0	217
	18	190	242	113	141	95	304	0	220
	19	35	180	16	198	216	167	0	90
	20	239	330	189	104	73	47	0	105
	21	31	346	32	81	261	188	0	137
	22	1	1	1	1	1	1	0	1
	23	0	0	0	0	0	0	0	0
1	0	2	76	303	141	179	77	22	96
	2	239	76	294	45	162	225	11	236
	3	117	73	27	151	223	96	124	136
	4	124	288	261	46	256	338	0	221
	5	71	144	161	119	160	268	10	128
	7	222	331	133	157	76	112	0	92
	8	104	331	4	133	202	302	0	172
	9	173	178	80	87	117	50	2	56
	11	220	295	129	206	109	167	16	11
	12	102	342	300	93	15	253	60	189
	14	109	217	76	79	72	334	0	95
	15	132	99	266	9	152	242	6	85
	16	142	354	72	118	158	257	30	153
	17	155	114	83	194	147	133	0	87
	19	255	331	260	31	156	9	168	163
	21	28	112	301	187	119	302	31	216
	22	0	0	0	0	0	0	105	0
	23	0	0	0	0	0	0	0	0
	24	0	0	0	0	0	0	0	0

RADDR	WADDR	WDATA	DATA(0)	DATA(1)	DATA(2)	RDATA
0	0	$O'\gamma(0)$	$O\gamma(0)$	$O\gamma(1)$	$O\gamma(2)$	—
2	1	$O'\gamma(1)$	$O'\gamma(0)$	$O\gamma(1)$	$O\gamma(2)$	$O\gamma(0)$
3	2	$O'\gamma(2)$	$O'\gamma(0)$	$O'\gamma(1)$	$O\gamma(2)$	$O\gamma(2)$

RADDR	WADDR	WDATA	DATA(0)	DATA(1)	DATA(2)	RDATA
0	0	$O'\gamma(0)$	$O\gamma(0)$	$O\gamma(1)$	$O\gamma(2)$	—
2	2	$O'\gamma(1)$	$O'\gamma(0)$	$O\gamma(1)$	$O\gamma(2)$	$O\gamma(0)$
3	3	$O'\gamma(2)$	$O'\gamma(0)$	$O\gamma(1)$	$O'\gamma(1)$	$O\gamma(2)$

In addition to reordering rows, it is also possible to **reorder columns**.

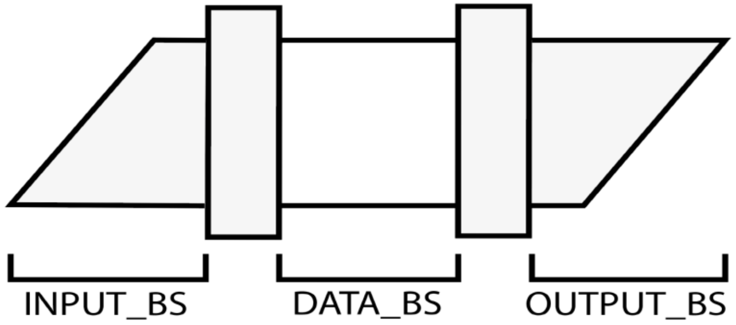
This optimization stems from the observation that two successive rows often have columns to process in common. For this reason, a **bypass** option has been added.

To take full advantage of this feature, it is necessary first to process the second phase within the PE, relating to the columns that are also present in the next row.

However, this **reordering of addresses inevitably affects the index's choice for the first phase**. It can no longer be chosen simply by increasing its value each time but must be congruent with that used for the second phase. In this way, the unread gamma values will not be overwritten.

BYPASS	INPUT_BS	DATA_BS	OUTPUT_BS
0	Oγ(0)[rot(0)]	—	—-
0	Oγ(1)[rot(1)]	Oγ(0)[rot(0)]	—
0	Oγ(2)[rot(2)]	Oγ(1)[rot(1)]	Oγ(0)[rot(0)]
0	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]	Oγ(1)[rot(1)]
1	Oγ(4)[rot(4)]	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]
1	Oγ(4)[rot(4)]	Oγ(4)[rot(4)]	Oγ(3)[rot(3)]
1	Oγ(4)[rot(4)]	Oγ(4)[rot(4)]	Oγ(4)[rot(4)]

BYPASS	CE	INPUT_BS	DATA_BS	OUTPUT_BS
0	1	Oγ(0)[rot(0)]	—	—-
0	1	Oγ(1)[rot(1)]	Oγ(0)[rot(0)]	—
0	1	Oγ(2)[rot(2)]	Oγ(1)[rot(1)]	Oγ(0)[rot(0)]
0	1	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]	Oγ(1)[rot(1)]
1	0	Oγ(4)[rot(4)]	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]
1	0	Oγ(4)[rot(4)]	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]
1	0	Oγ(4)[rot(4)]	Oγ(3)[rot(3)]	Oγ(2)[rot(2)]



A final consideration should be made when it is necessary or optional to pipeline the barrel shifter.

In this case, it will be necessary to implement a **chip enable** to *block the data rotation process during the bypass period.*

Without a chip enable, all the data inside the barrel shifter would inevitably be lost during the bypass period, as can be seen in the table on the left.

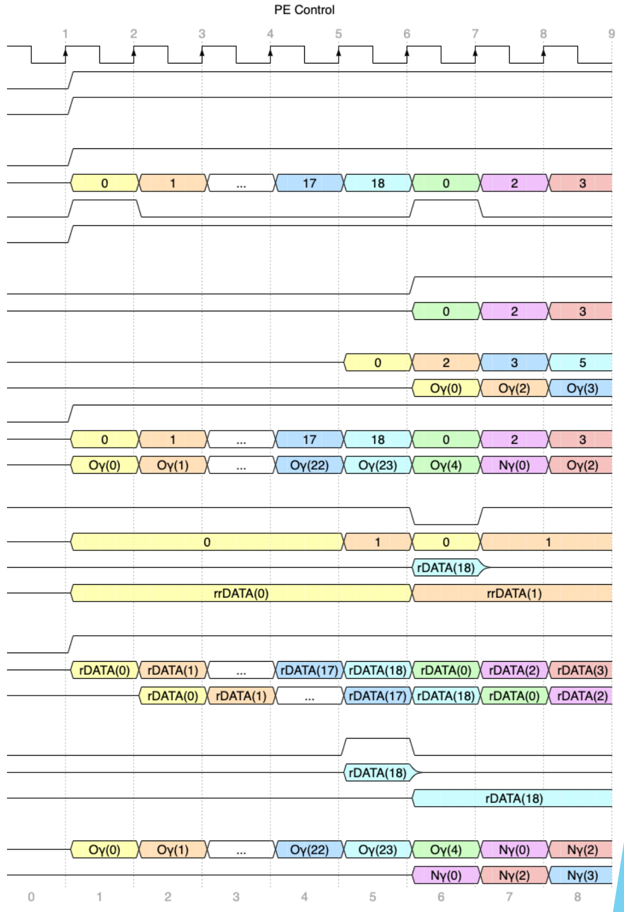
PE Timing Diagarm

This timing diagram shows PE processing in the case of the two rows of the base graph.

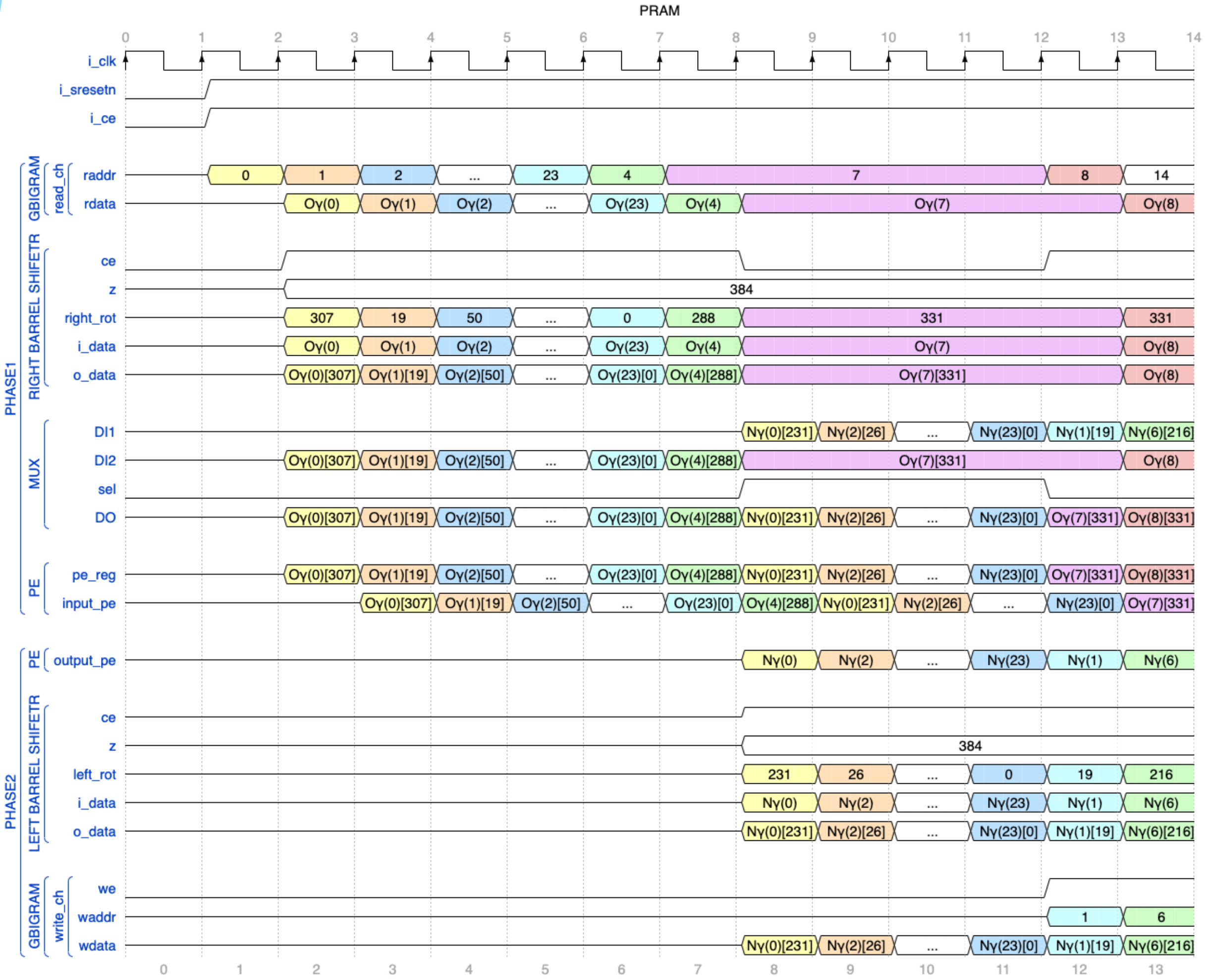
The values highlighted in yellow represent the columns present in both rows.

The values highlighted in orange represent the rotation value associated with each column.

0	0	250	307	73	223	211	294	0	135
	1	69	19	15	16	198	118	0	227
	2	226	50	103	94	188	167	0	126
	3	159	369	49	91	186	330	0	134
	5	100	181	240	74	219	207	0	84
	6	10	216	39	10	4	165	0	83
	9	59	317	15	0	29	243	0	53
	10	229	288	162	205	144	250	0	225
	11	110	109	215	216	116	1	0	205
	12	191	17	164	21	216	339	0	128
	13	9	357	133	215	115	201	0	75
	15	195	215	298	14	233	53	0	135
	16	23	106	110	70	144	347	0	217
	18	190	242	113	141	95	304	0	220
	19	35	180	16	198	216	167	0	90
	20	239	330	189	104	73	47	0	105
	21	31	346	32	81	261	188	0	137
	22	1	1	1	1	1	1	0	1
	23	0	0	0	0	0	0	0	0
1	0	2	76	303	141	179	77	22	96
	2	239	76	294	45	162	225	11	236
	3	117	73	27	151	223	96	124	136
	4	124	288	261	46	256	338	0	221
	5	71	144	161	119	160	268	10	128
	7	222	331	133	157	76	112	0	92
	8	104	331	4	133	202	302	0	172
	9	173	178	80	87	117	50	2	56
	11	220	295	129	206	109	167	16	11
	12	102	342	300	93	15	253	60	189
	14	109	217	76	79	72	334	0	95
	15	132	99	266	9	152	242	6	85
	16	142	354	72	118	158	257	30	153
	17	155	114	83	194	147	133	0	87
	19	255	331	260	31	156	9	168	163
	21	28	112	301	187	119	302	31	216
	22	0	0	0	0	0	0	105	0
	23	0	0	0	0	0	0	0	0
	24	0	0	0	0	0	0	0	0



PRAM Timing Diagram



0	0	250	307
	1	69	19
	2	226	50
	3	159	369
	5	100	181
	6	10	216
	9	59	317
	10	229	288
	11	110	109
	12	191	17
	13	9	357
	15	195	215
	16	23	106
	18	190	242
	19	35	180
	20	239	330
	21	31	346
	22	1	1
	23	0	0
1	0	2	76
	2	239	76
	3	117	73
	4	124	288
	5	71	144
	7	222	331
	8	104	331
	9	173	178
	11	220	295
	12	102	342
	14	109	217
	15	132	99
	16	142	354
	17	155	114
	19	255	331
	21	28	112
	22	0	0
	23	0	0
	24	0	0

QUESTIONS?

THANK YOU FOR YOUR ATTENTION