Input Ports

clk

Width : 1
Description :
system clock

rst_n

Width : 1
Description :
Set to 0 to reset all memory blocks (pre-written memory blocks are not affected). This will override all other signals.

write_en

Width : 1
Description :
Set to 1 to write data to a currently available memory unit (you should write data to data_in simultaneously).

One memory unit can only be written once, and can hold at most MAX_DEPTH (currently 1024) notes or MAX_DEPTH (currently 1024) of samples. The memory block will automatically stop writing when it is full, and will switch to another memory unit once write_en is set to 0.

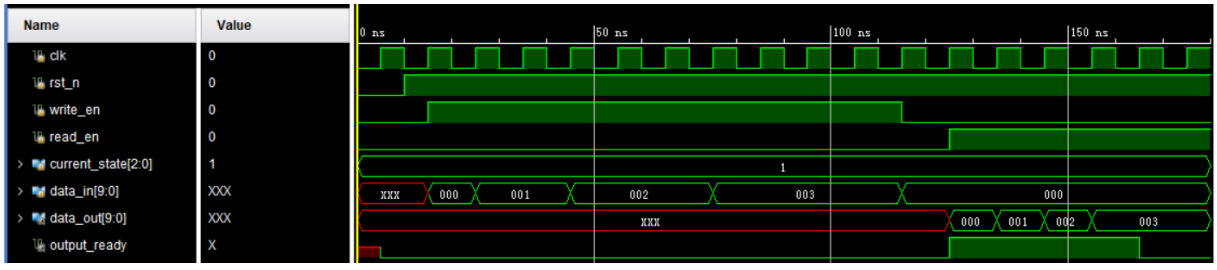
Before you write, check full_flag to see if memory is full.

read_en

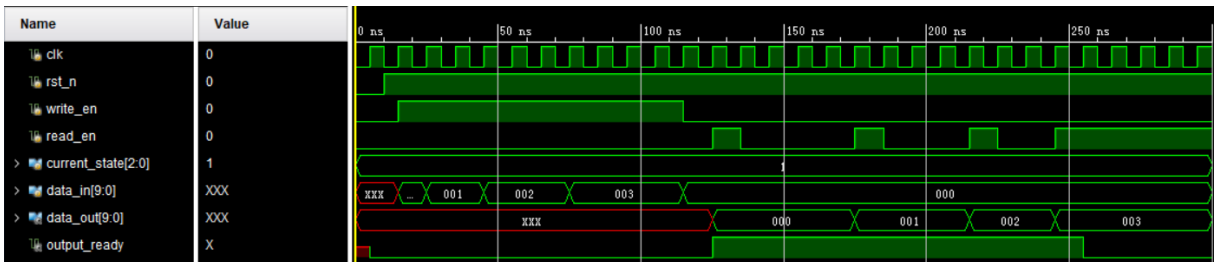
Width : 1
Description :
Set to 1 to read data from selected memory unit (data_out will begin simultaneously, if the memory unit is not empty). When read_en is set to 0, data_out will tend to hold the last value it has (you can verify whether the data is still valid by checking output_ready).

Only in learning state (state parameter LEARNING, currently 1) or auto state (state parameter AUTOPLAY, currently 0) can you read data. Otherwise, the memory block will ignore read_en.

In learning state, for each positive edge of clk, if read_en is set to 1, the memory block will output one note to data_out for 1 clk cycle, regardless of the duration of the note when it is written.

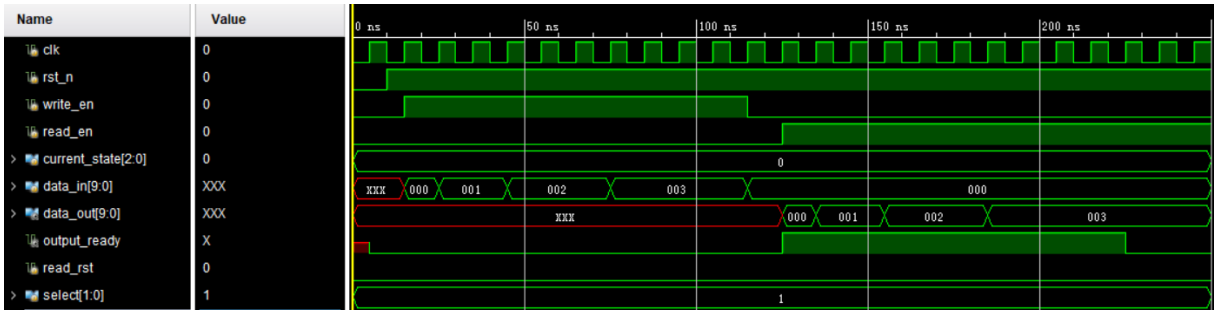


Example: LEARNING state with continuous read_en

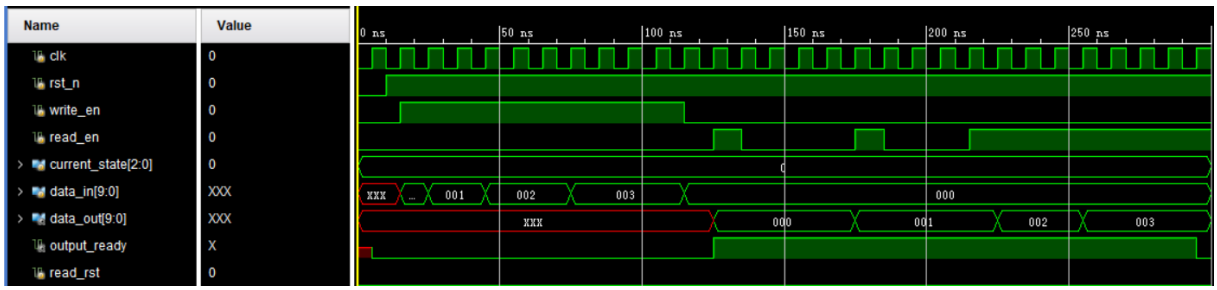


Example: LEARNING state with discrete read_en

In auto state, for each positive edge of clk, if read_en is set to 1, the memory block will output one sample to data_out. The time duration of each note is determined by the duration of the note when it is written. (Note that in figure 2 signal 001 still holds for 2 clk cycles, if you concatenate the part where read_en is 1)



Example: AUTOPLAY state with continuous read_en



Example: AUTOPLAY state with discrete read_en

You should always set read_en to 1 until output_ready is 0 for auto state. Otherwise, the output data will not reflect the actual duration of each note (like in figure 2).

read_rst

Width : 1
Description :
Set to 1 to reset the read pointer of all memory unit to the beginning of the memory unit. This will override all other signals, except rst_n. It is recommended to set read_rst to 1 before every read operation.

current_state

Width : STATE_WIDTH (currently 3)
Description :
current state of the controller
000 : AUTOPLAY
001 : LEARNING
others: reserved

select

Width : MAX_MEMORY_BIT (currently 2)
Description :
Select which memory unit to read from (there are MAX_MEMORY, currently 4, memory units in total). However, you don't need to set select when you are writing data to memory unit, because the memory block will automatically select the first available memory unit to write to.
00 : reserved for pre-written memory unit, any write operation will be ignored.

data_in

Width : DATA_WIDTH (currently 10)
Description :
Data to be written to memory unit. The first 8 bits for note, the last 2 bits for octave. (an assignment should look like: assign data_in = {note, octave};)

Output Ports

data_out

Width : DATA_WIDTH (currently 10)
Description :
Data read from memory unit (will start simultaneously with read_en). The first 8 bits for note, the last 2 bits for octave. (an assignment should look like: assign {note, octave} = data_out;)

name_info

Unfinished. Current width is 1.

output_ready

Width : 1

Description :

Set to 1 when data_out is valid and ready to be read (will start simultaneously with read_en).

Any output data in data_out should be ignored if output_ready is 0. Normally, it will only be 0 when read operation is complete, or before the first read operation. But a low rst_n or a high read_rst will also set output_ready to 0.

full_flag

Width : 1

Description :

Set to 1 when the all memory units are full.

Any write operation to a full memory unit will be ignored. A low rst_n will set full_flag to 0.

count

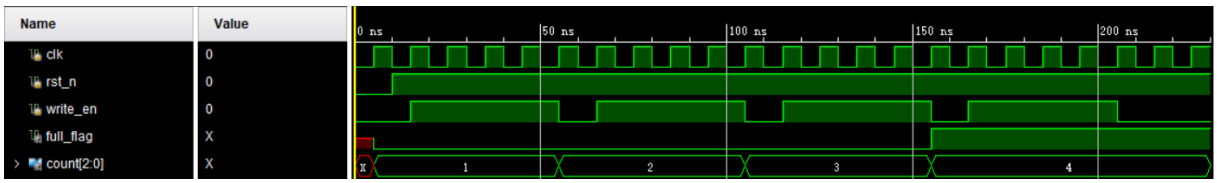
Width : MAX_MEMORY_BIT (currently 2) + 1

Description :

The number of memory units that are full.

This is useful when you want to know how many memory units are full, and to know which memory units can be read from.

Default value is PRE_WRITTEN_COUNT (currently 1), which is the number of pre-written memory units. It will be updated on negtive edge of write_en. A low rst_n will set count to PRE_WRITTEN_COUNT.



Example: full_flag and count

Parameters

Parameter	Default Value	Description	In Bits or Units
DATA_WIDTH	10	Width of data_in and data_out	bits
MAX_DEPTH	1024	Maximum number of notes or samples in each memory unit	units
MAX_DEPTH_BIT	10	Number of bits needed to represent MAX_DEPTH	bits
SAMPLE_INTERVAL	50000000	Interval between samples (unit: clk cycle)	units
MAX_SAMPLE_INTERVAL	32	Maximum interval (2**MAX_SAMPLE_INTERVAL) between samples	bits
MAX_MEMORY	4	Number of memory units	units
MAX_MEMORY_BIT	2	Number of bits needed to represent MAX_MEMORY	bits
STATE_WIDTH	3	Number of bits needed to represent state	bits
PRE_WRITTEN_COUNT	1	Number of pre-written memory units	units
AUTOPLAY	0	AUTOPLAY state	units
LEARNING	1	LEARNING state	units