ARM memory generator

Arm Memory generator

- Make sure you create a folder similar to what you did for other hws. Then for memory generator point to the location that the executable arm memory generator command can exectute.
- For example:

  Single port register file:

  /afs/umbc.edu/depts/cmpe/vlsi/cmpe641_sp16/ARM/arm/csm/ch013n/rf_sp_hdd_rvt_rvt/r4p0-00eac0/bin

  Single port SRAM:

  /afs/umbc.edu/depts/cmpe/vlsi/cmpe641_sp16/ARM/arm/csm/ch013n/sram_sp_hdf_rvt_rvt/r5p0-01eac0/bin

  Executable file: sram_sp_hdf_rvt_rvt

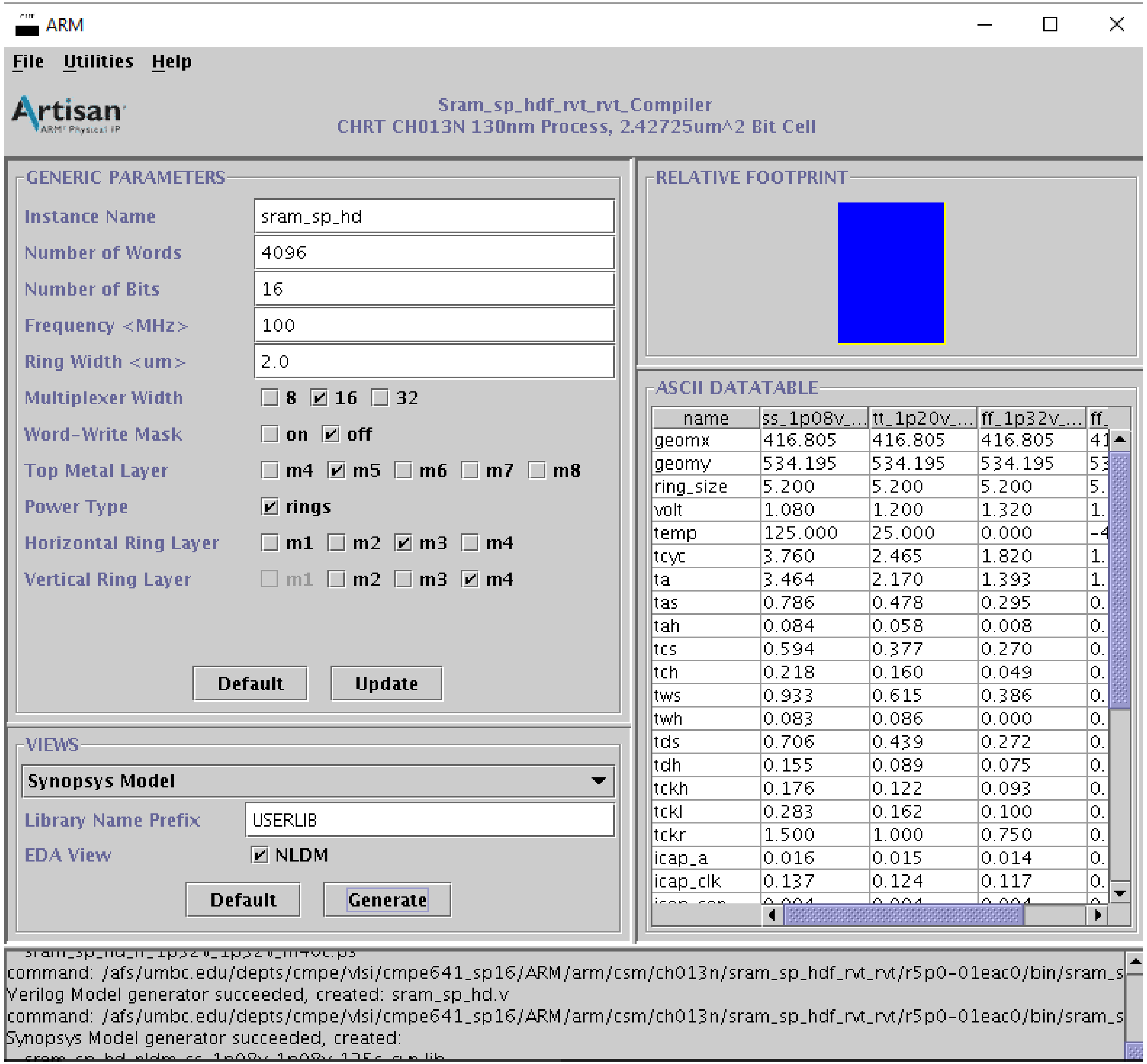
- So basically in your Verilog folder you execute:

```
/afs/umbc.edu/depts/cmpe/vlsi/cmpe641_sp16/ARM/arm/csm/ch013n/sram_sp_hdf_rvt_rvt/r5p0-01eac0/bin/sram_sp_hdf_rvt_rvt
```

- When you execute the command then GUI ARM memory generate will showup as long as you are able to execute GUI in your server.

Screen shot for ARM Memory generator

- Every time that you change any parameter you need to Update.
- For Views, you should choose Verilog model, Synopsis Model, to generate separately.



Sram_sp_hdf_rvt_rvt_Compiler
CHRT CH013N 130nm Process, 2.42725um^2 Bit Cell

GENERIC PARAMETERS

Instance Name: sram_sp_hd

Number of Words: 4096

Number of Bits: 16

Frequency <MHz>: 100

Ring Width <um>: 2.0

Multiplexer Width: □ 8 ☑ 16 □ 32

Word-Write Mask: □ on ☑ off

Top Metal Layer: □ m4 ☑ m5 □ m6 □ m7 □ m8

Power Type: ☑ rings

Horizontal Ring Layer: □ m1 □ m2 ☑ m3 □ m4

Vertical Ring Layer: □ m1 □ m2 □ m3 ☑ m4

VIEWS

Synopsys Model

Library Name Prefix: USERLIB

EDA View: ☑ NLDM

ASCII DATATABLE

name	ss_1p08v_...	tt_1p20v_...	ff_1p32v_...	ff_
geomx	416.805	416.805	416.805	41
geomy	534.195	534.195	534.195	53
ring_size	5.200	5.200	5.200	5.
volt	1.080	1.200	1.320	1.
temp	125.000	25.000	0.000	-4
tcyc	3.760	2.465	1.820	1.
ta	3.464	2.170	1.393	1.
tas	0.786	0.478	0.295	0.
tah	0.084	0.058	0.008	0.
tcs	0.594	0.377	0.270	0.
tch	0.218	0.160	0.049	0.
tws	0.933	0.615	0.386	0.
twh	0.083	0.086	0.000	0.
tds	0.706	0.439	0.272	0.
tdh	0.155	0.089	0.075	0.
tckh	0.176	0.122	0.093	0.
tckl	0.283	0.162	0.100	0.
tckr	1.500	1.000	0.750	0.
icap_a	0.016	0.015	0.014	0.
icap_clk	0.137	0.124	0.117	0.
icap_cen	0.004	0.004	0.004	0.

```
  sram_sp_hd_ff_1p32v_1p32v_m40c.ps
command: /afs/umbc.edu/depts/cmpe/vlsi/cmpe641_sp16/ARM/arm/csm/ch013n/sram_sp_hdf_rvt_rvt/r5p0-01eac0/bin/sram_s
Verilog Model generator succeeded, created: sram_sp_hd.v
command: /afs/umbc.edu/depts/cmpe/vlsi/cmpe641_sp16/ARM/arm/csm/ch013n/sram_sp_hdf_rvt_rvt/r5p0-01eac0/bin/sram_s
Synopsys Model generator succeeded, created:
  sram_sp_hd_nldm_ss_1p08v_1p08v_125c_syn.lib
```

- Sample generated files in your folder:

```
linuxserver1.cs.umbc.edu[188] cd verilog/
linuxserver1.cs.umbc.edu[189] ls
ACI.log                    sram_sp_hd_nldm_ff_1p32v_1p32v_0c_syn.lib
sram_sp_hd_nldm_tt_1p20v_1p20v_25c_syn.lib  sram_sp_hd.v
sram_sp_hd_ff_1p32v_1p32v_0c.ps
sram_sp_hd_nldm_ff_1p32v_1p32v_m40c_syn.lib
sram_sp_hd_ss_1p08v_1p08v_125c.ps
sram_sp_hd_ff_1p32v_1p32v_m40c.ps
sram_sp_hd_nldm_ss_1p08v_1p08v_125c_syn.lib
sram_sp_hd_tt_1p20v_1p20v_25c.ps
```

- For both simulation and implementation, you instantiate the memory that you generated in your top Verilog file that you write.

Modifying rc script file for synthesis

- For synthesis in rc script that you have
  - Make sure you point to the location of generated Verilog files for the memory in addition to your other Verilog files
    - set_attribute hdl_search_path
  - Make sure you point to the location of .lib file that was generated for your memory
    - set_attribute lib_search_path
  - Make sure you put the exact library name
    - set_attribute library
  - List your verilog files but NOT the Verilog file that was generated
    - Ex: set myFiles [list top.v] ;