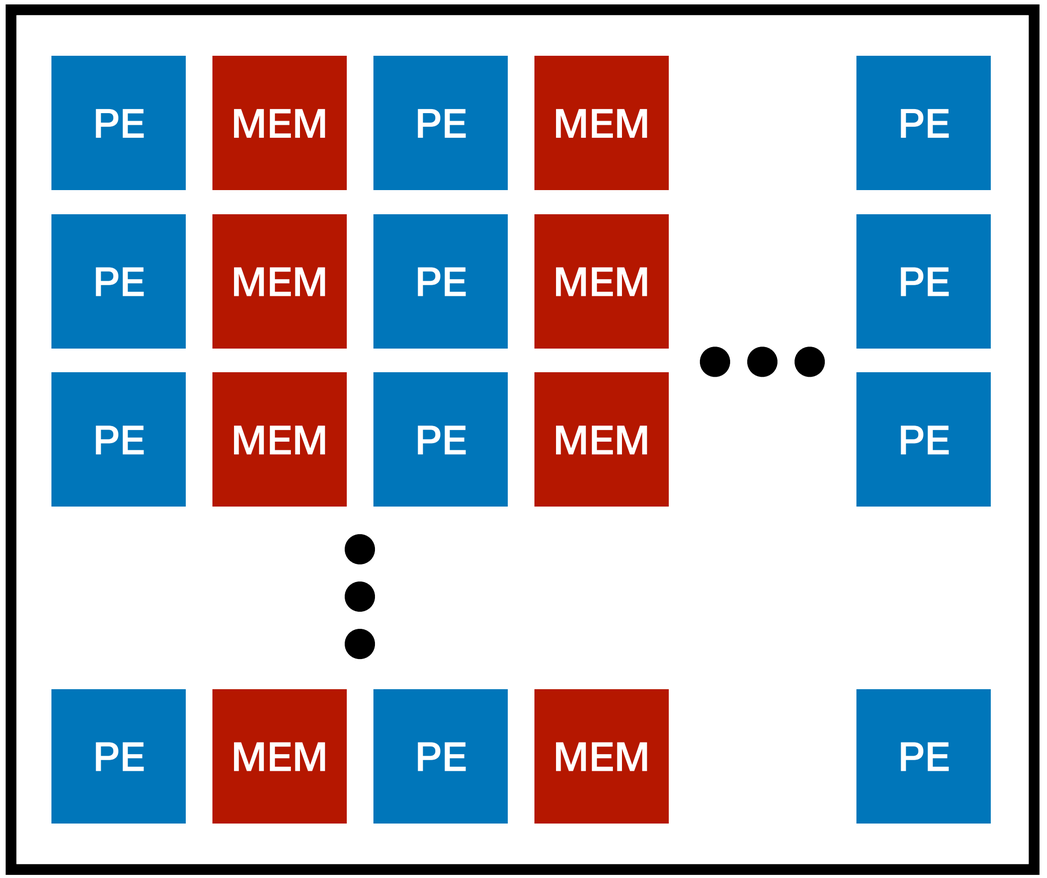
Generators: what we did

Genesis2 (Perl + Verilog)

```
//; for (my $h=0; $h<$cgra_grid_height; $h++) {
//;   for (my $w=0; $w<$cgra_grid_width; $w++) {
//;     my $tile_type = $tile_grid{$key};
//;     if ($tile_type eq "mem") {
//;       my $data_bus = $tile_config->
//;           { $tile_type }{ 'gen_mem_for_busname' };
//;       my $bus_width = $bus_width_hash { $data_bus };
   wire [`$bus_width-1`:0] mem_chain_`$h`_`$w`;
   wire  mem_chain_valid_`$h`_`$w`;
//;     }
//;     if (($tile_type eq "mem") || ($tile_type eq "pe")) {
//;       for (my $i=0; $i<$global_signal_count; $i++) {
//;         if ((${w}%2==0) && (${h}%2==0)) {
   wire global_wire_h2l_1_`${i}`_`${w}`_`${h}`;
//;         }
//;       }
   wire global_wire_l2h_0_`${w}`_`${h}`;
//;     }
//;   }
//; }
```

“Sea of Tiles”



Generators: what we want

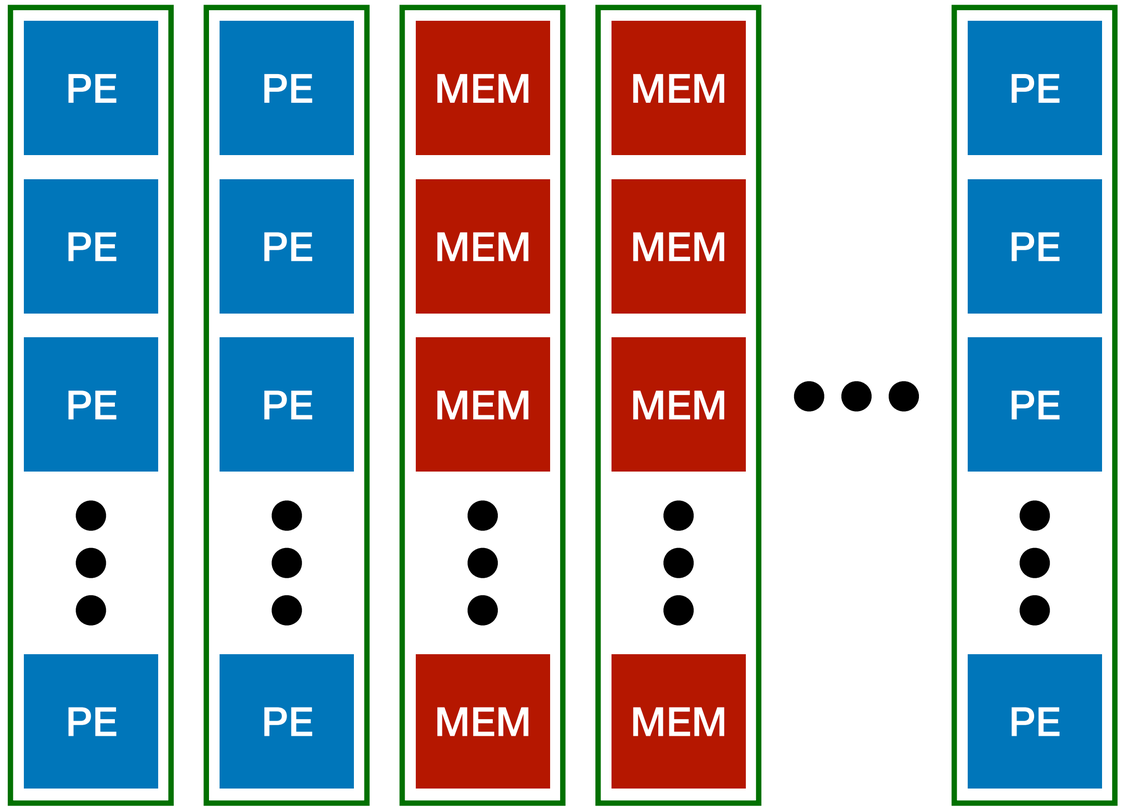
- Move away from simple text replacement towards high-level languages
    - This gives us more semantic information
- “Unflatten” the design - more hierarchy aids development and **physical design**
- Have a single source of truth for design, verif., and programming
- Move incrementally: new generators should interoperate w/ old

Generators: what we want

Python

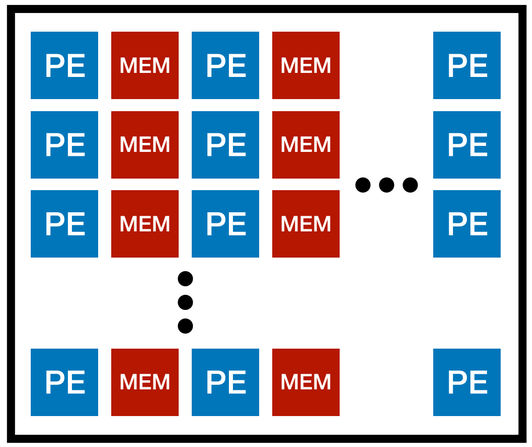
```
import pe, memory

def create_column(height, ...):
    pes = [pe(...) for _ in range(height)]
    mems = [mem(...) for _ in range(height)]
    for i in range(height):
        wire(pes[i].east, mems[i].west)
        wire(pes[i].south, pes[i + 1].north)
    return column(pes, mems)

def cgra(col_height, num_cols, ...):
    cols = []
    for i in range(num_cols):
        cols.append(create_column(col_height, ...))
        wire(cols[i - 1], cols[i])

if __name__ == "__main__":
    col_height, num_cols, ... = get_opts()
    my_cgra = cgra(col_height, num_cols, ...)
    my_cgra.generate_verilog()
```

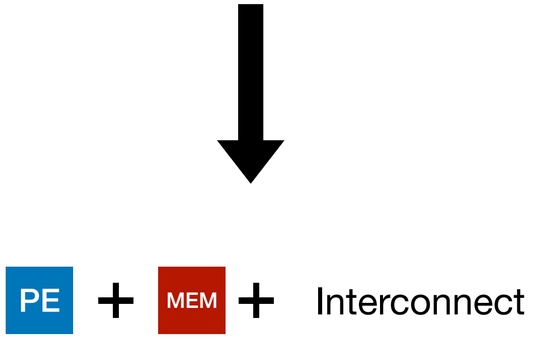
Column Layout

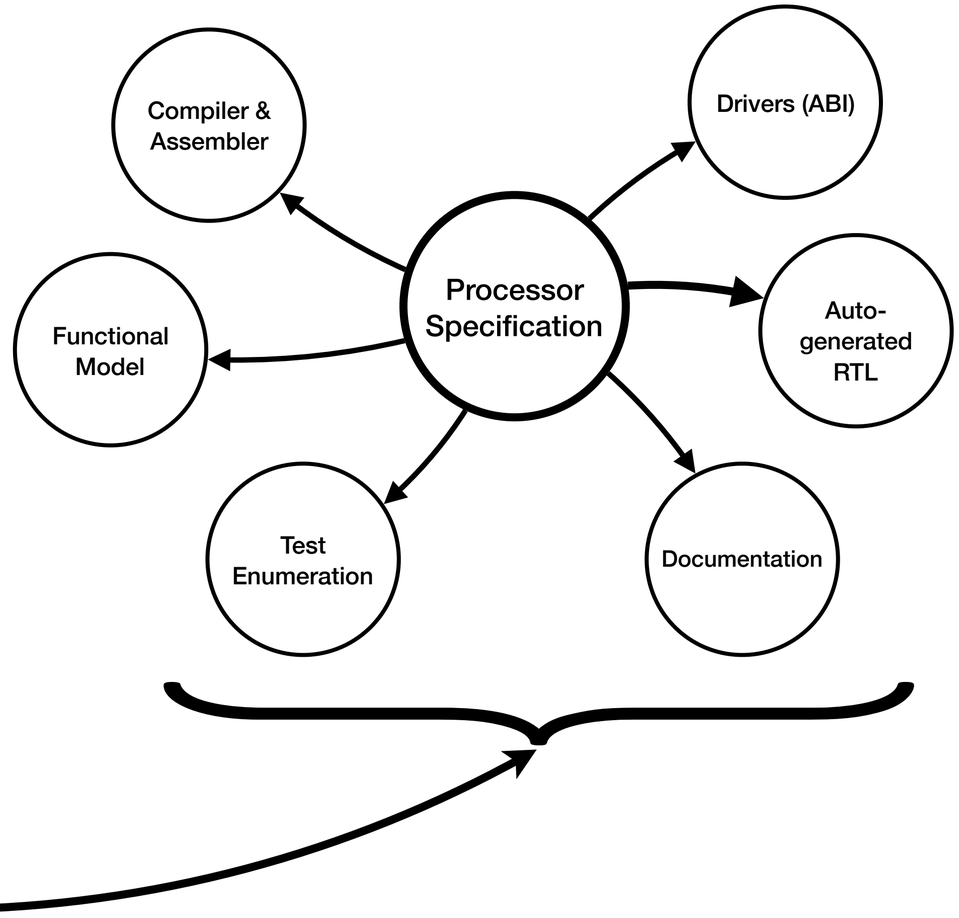


Generators: what we want

- Specify PE's and memories separately
- Derive all collateral from single source-of-truth
- Wire up PE's and memories together to construct final CGRA design







Generators: what we want

- Use Generators for design space exploration (DSE)
- Well designed generators enable easy and efficient DSE
    - Requires intuitive parameters
    - What knobs should we provide?
- Actually need to sweep the parameter space! (didn’t do this)
- Create a framework that allows for fast, quantitative evaluation of many different parameterizations for given applications