Extending gem5 with DRAMSim and DRAMSys

Ports are useful interfaces to other simulators

Why use an external simulator?

> Note: I don't advise using external DRAM simulators.
> gem5's DRAM model is accurate enough for most research

The main reasons to use an external DRAM simulator are:

- For comparisons between gem5's DRAM models and other simulators (e.g., when developing a new DRAM model for gem5)
- When you have already modified the other simulator and need to drive it with realistic traffic

Getting DRAMSys

See `gem5/ext/dramsys/README` for deatils.

Run

```
cd ext/dramsys
git clone https://github.com/tukl-msd/DRAMSys --branch v5.0 --depth 1 DRAMSys
```

Buidling DRAMSys

After you add the DRAMSys repo to `ext/dramsys`, it will automatically be built into gem5.

```
scons build/NULL/gem5.opt -j$(nproc)
```

Using DRAMSys

See https://github.com/tukl-msd/DRAMSys for documentation on DRAMSys

To configure gem5 to use DRAMSys, you can use the standard library.
DRAMSys can be used as a `MemorySystem` just like the `SingleChannel` or `MultiChannel` memories.

Open `materials/05-Other-simulators/02-dram/dramsys-example.py`.

Add the following lines to create a memory system with DDR4 from DRAMSys

```
memory = DRAMSysMem(
    configuration="/workspaces/2024/gem5/ext/dramsys/DRAMSys/configs/ddr4-example.json",
    recordable=True,
    resource_directory="/workspaces/2024/gem5/ext/dramsys/DRAMSys/configs",
    size="4GB",
)
```

Configuring DRAMSys

Options for DRAMSys:

- `configuration`: See `gem5/ext/dramsys/DRAMSys/configs/` for the provided configurations.
  - Must be absolute or relative to your run path.
- `resource_directory`: Pointer to the configs directory.
  - Must be absolute or relative to your run path.
- `recordable`: Whether DRAMSys should record a trace file

Note on implementation

- DRAMSys uses TLM 2.0
- This is a good example of how to get gem5 to talk to a TLM object.

DRAMSys output

```
../../../gem5/build/NULL/gem5.opt dramsys-example.py
```

```
board.memory.dramsys.DRAMSys.controller0  Total Time:     250027920 ps
board.memory.dramsys.DRAMSys.controller0  AVG BW:          87.97 Gb/s |  11.00 GB/s |  73.67 %
board.memory.dramsys.DRAMSys.controller0  AVG BW\IDLE:     87.97 Gb/s |  11.00 GB/s |  73.67 %
board.memory.dramsys.DRAMSys.controller0  MAX BW:         119.40 Gb/s |  14.93 GB/s | 100.00 %
```

Outputs a file, `board.memory.dramsys.DRAMSys_ddr4-example_example_ch0.tdb` which is a database trace.

> Does not use gem5's stats output!

DRAMSim

Similar to DRAMSys in how to obtain and use

> Note: DRAMSim3 is not tested regularly

See `gem5/ext/dramsim3/README` for details.

To get DRAMsim run the code below and recompile gem5.

```
cd ext/dramsim3
git clone clone git@github.com:umd-memsys/DRAMsim3
cd DRAMsim3
mkdir build
cd build
cmake ..
make -j$(nproc)
```

Using DRAMSim3

In `gem5.components.memory.dramsim_3` there are single channel configurations available.

```
from gem5.components.memory.dramsim_3 import SingleChannelHBM
```

```
memory = SingleChannelHBM(size="1GiB")
```

You can extend this with other memories using `SingleChannel` from that module.

Find the list of DRAMSim configs: `gem5/ext/dramsim3/DRAMsim3/configs`

```
SingleChannel(<memory type from configs>, <size>)
```

NOTE: DRAMSim3 doesn't work with v24.0.0.0!