Hardware Garbage Collection

Martha Barker (mmb2295)
CSEE 4840: Embedded System Design
Spring 2019
Architecture

- Userspace Op Generator
- Userspace Reference Program
- Memory
  - Garbage Collector
Communication

Read/Write interface over Avalon bus

Hardware unit is slave device

32 bit data

4 bit address

8 Addresses:

1. Status
2. Num Roots
3. Base
4. Bound
5. GC Threshold
6. Read Op/Res
7. Write Op/REs
8. BRAM
Hardware
Status Register

Bitmap:

- 0: Ready for read op
- 1: ready for write op
- 2: read response valid
- 3: write response valid
- 4: need gc
- 5: gc complete
- 6: ready for root input
Handshake Protocol

<table>
<thead>
<tr>
<th>valid</th>
<th>ready</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Token valid; not consumed</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Token transferred</td>
</tr>
<tr>
<td>0</td>
<td>–</td>
<td>No token to transfer</td>
</tr>
</tbody>
</table>

Hardware expects handshake protocol

Ready signal won’t assert high for inputs until there is valid data at its input

Status register shows ready for input whenever input is consumed then holds valid input until hardware accepts token
Software

User defined parameters:

- Memory base
- Memory bound
- GC threshold
- Num Roots
- Num Ops
- Op distribution (read/write)

Main.c

Operational loop

Polls hardware for status

Writes input ops and reads output results to hardware and reference implementation

Compares outputs and BRAM contents
Software

Reference.c

Keeps a reference struct containing BRAM and list of roots

Write to the reference and read from the reference exactly as hardware should

Do garbage collection on reference

Roots in list have a “lifetime” that can be decremented

Hardware.c

Helper methods to access the module driver

- Initialize memory
- Poll status
- read/write ops
- read/write results
- Root ops
Validation

Main software loop checks all read and write results read from hardware against reference implementation.

Program prints error and exits if hardware does not match reference.

Read out BRAM and compare to reference before and after garbage collection.

Program prints error if hardware BRAM does not match reference.
Limitations

Need to reboot hardware after every run

To reset I would need to clear every token from the hardware system

- Significant changes to generated code