Cache FSM

Pritesh Kalantri, Keifer Caebe

Legend:
Rd & Wr = ERR state
Tag_in = Instr[15:11]
Index = Instr[10:3]
Offset = Instr[2:0]
Data_in = Data_in
Addr = Addr

Wait_Req

~done, ~stall

Check_Rd

Cache: Enable, Index, Tag_in, Offset, Comp
System: Stall = ~done
Dataout = (done) ? cacheout : 0,
done = hit & valid

READ_MEM_1

Memory: Rd, Addr = {AddrIn[15:3]3'b000}
System: ~done, stall

READ_MEM_2

Memory: Rd, Addr = {AddrIn[15:3]3'b010}
System: ~done, stall

READ_MEM_3

Memory: Rd, Addr = {AddrIn[15:3]3'b100}
Cache: Enable, Write, offset = 0, tag_in,
data_in, ~valid_in
System: ~done, stall

READ_MEM_4

Memory: Rd, Addr = {AddrIn[15:3]3'b110}
Cache: Enable, Write, offset = 1, tag_in,
data_in, ~valid_in
System: ~done, stall

LOAD_CACHE_3

Cache: Enable, Write, offset = 2, tag_in,
data_in, ~valid_in
System: ~done, stall

LOAD_CACHE_4

Cache: Enable, Write, offset = 3, tag_in,
data_in, valid_in
System: done = rd, stall = wr

Error
----
err

Cache_Write

Cache: Enable, Index, Tag_in, Offset, Comp,
write, data_in
System: Stall = ~done
Dataout = (done) ? cacheout : 0,
done = hit & valid

Done

Done = 1
Stall = 0
DataOut = (Rd) ? CacheOut : 0

Cache_Write

Cache: Enable, Write, valid_in, Index, Offset,
Tag_in, Data_in
System: Done, ~stall

Cache_Writeback_1

Mem:Wr, data_in, addr = {tagOut, index, off}
Cache: Enable, Index, Offset = 0
System: stall, ~done

Cache_Writeback_2

Mem: Wr, data_in, addr={tagOut, index, off}
Cache: Enable, Index, Offset = 1
System: stall, ~done

Cache_Writeback_3

Mem: Wr, data_in, addr={tagOut,index,off}
Cache: Enable, Index, Offset = 2
System: stall, ~done

Cache_Writeback_4

Mem: Wr, data_in, addr={tagOut,index,off}
Cache: Enable, Index, Offset = 3
System: stall, ~done

Transitions:
- ~rd, ~wr
- rst
- rd, ~wr
- rd, wr
- hit & valid & Rd & ~Wr
- ~valid | (valid & ~dirty)
- hit & valid & ~Rd & ~Wr
- ~hit & valid & dirty
- hit & valid & ~Rd & Wr
- Rd & ~Wr
- ~Rd & Wr
- ~rd, wr
- hit & valid
- ~hit & valid & dirty
- ~hit | (hit & ~valid)
- Rd
- Wr