Thinking about Transactional Memory, Axiomatically

Nathan Chong
ARM Research (Security and Correctness)

ARM v8.0 Axiomatic Model by Will Deacon (ARM)
TM work with John Wickerson and Tyler Sorensen (Imperial)
Multiprocessor program

\[ T_0 \quad T_1 \quad \ldots \quad T_{n-1} \]

a sequence of assembly instructions
(ADD, CMP, B, LDR, STR, …)

Essential Question: What values can this load return?
Stronger

Weaker

Sequential Consistency

x86 (TSO)

Alpha

Power

ARM

RISC-V
Sequential Consistency

- Alpha
- Power
- ARM
- ARM v8.0
- x86 (TSO)

Stronger → Weaker
Stronger

Weaker

Sequential Consistency

x86 (TSO)

Alpha

Power

ARM

ARM v8.0

RISC-V

TM Design Space

Sequential Consistency
This Talk

- ARM’s new, stronger v8.0 memory model
  - Multi-copy atomicity
  - Store token optimisation (disallowed)
  - False data and control dependencies (enforced)

- Extending weak memory with transactional memory
Part 1: The ARM v8.0 Memory Model

Work by Will Deacon (ARM)
The set of all executions

ARM v8.0
The v8.0 Memory Model is Multi-Copy Atomic
a: $W[x]=1$
a: \( W[x] = 1 \) \hspace{1cm} \text{rfe} \hspace{1cm} b: \ R[x] = 1
a: $W[x]=1$ \hspace{1cm} b: $R[x]=1$ \hspace{1cm} c: $R[x]=?$

*Parameterised by memory-model*
a: \( W[x] = 1 \) \\

*Parameterised by memory-model* \\

b: \( R[x] = 1 \) \\

c: \( R[x] = ? \)

(c) can read 0 => Non-MCA (e.g, ARMv7, Power, RISC-V) \\
(c) must read 1 => MCA (e.g, ARMv8, x86)
a: $W[x]=1$ \hspace{1cm} rfe \hspace{1cm} b: $R[x]=1$

happens-before

$R[x]=0$
WRC+addrs

a: \( W[x]=1 \)

b: \( R[x]=1 \)

c: \( W[y]=1 \)

d: \( R[y]=1 \)

e: \( R[x]=0 \)

Prev: Allowed / v8.0: Disallowed
IRIW+addr

a: $W[x]=1$ → b: $R[x]=1$

c: $R[y]=0$

d: $W[y]=1$ → e: $R[y]=1$

f: $R[x]=0$

Prev: Allowed / v8.0: Disallowed
IWP2.4+addrs

Prev: Allowed / v8.0: Allowed

a: \( W[x] = 1 \)

d: \( W[y] = 1 \)

b: \( R[x] = 1 \)

e: \( R[y] = 1 \)

c: \( R[y] = 0 \)

f: \( R[x] = 0 \)
Part 2: Transactional Memory Extension

Joint work with John Wickerson and Tyler Sorensen (Imperial)
[HM93] A transaction is a finite sequence of machine instructions, executed by a single process, satisfying the following properties:

- Serializability
- Atomicity (All or Nothing)
Shared Memory

Tx₁

Tx₂
Shared Memory

Tx1

Tx2

conflict!
Shared Memory

Tx₁

Replay Tx₁
Transactional Memory Architectural Changes

ISA Changes: TXSTART, TXCOMMIT, TXABORT, TXTEST

Impact on Memory Model

Interaction with exceptions, virtualisation, SVE, debug, ...
An Execution is a Structure

Execution $X = \langle E, R, W, F, po, data, ctrl, addr, rf, co, \ldots \rangle$

- a set of events
- basic sets
- basic relations
Introduce a Same-Transaction Relation

Execution $X = \langle E, R, W, F, po, data, ctrl, addr, rf, co, …, \text{stxn} \rangle$

an equivalence relation identifying events of the same (successfully committed) transaction

Define $T = \text{domain}(\text{stxn})$ // the set of transactional events
stxn

is equivalent to

po

po

po
stxn Must Be Contiguous in po

(Power TM supports transaction pausing)
Can’t Express Some Instruction Sequences

TXSTART
[...]
TXCOMMIT

// Nested
TXSTART
TXSTART
[...]
TXCOMMIT
TXCOMMIT

// No commit
TXSTART
[...]
TXCOMMIT
[...]

// Unbalanced
TXSTART

Ill-defined program?
A Transactional Variant of MP

Disallow (transaction serialisation)

acyclic stxn; (rfe | coe | fre)
A Transactional Variant of MP

Disallow (transaction serialisation)

acyclic stxn; (rfe | coe | fre)
A Transactional Variant of MP

Disallow (transaction serialisation)

acyclic stxn; (rfe | coe | fre)
A Transactional Variant of MP

a: $W[x]=1$

b: $W[y]=1$

c: $R[y]=1$

d: $R[x]=0$

Disallow (transaction serialisation)

acyclic \textit{stxn}; (rfe | coe | fre)
A Transactional Variant of MP

Disallow (transaction serialisation)

acyclic stxn; (rfe | coe | fre)
Swapping Writes of MP

Disallow (transaction serialisation)

acyclic stxn; (rfe | coe | fre)
Interaction with Non-Transactional Events

Disallow => Strong Isolation [MBL06]

acyclic stxn; (rfe | coe | fre)+
The set of all executions

N is stronger-than M
M is weaker-than N
Finding executions in this set M\N is a SAT problem [WBS+17]
Weak Isolation \ Strong Isolation (3ev Solutions)

acyclic stxn; (rfe | coe | fre) as WeakIsolation
acyclic stxn; (rfe | coe | fre)+ as StrongIsolation

\begin{align*}
\text{txRR} & \quad \text{(Non-Interference)} \\
\text{txRW} \\
\text{txWR} \\
\text{txWW} & \quad \text{(Containment)}
\end{align*}
Transactional Sequential Consistency

“memory accesses from a given transaction should be contiguous in the total execution order” [DS09]

Global switch cannot change when executing a transaction
Current Landscape

- ARM
  - ARMv8.0
    - ARMv8.0+TM
      - SC
      - TSC
Future Work

- Exploring design space of TM with weak memory
  - Intra-thread ordering, Empty txs, Failing txs, MCA txs, ...
  - Operational modelling
  - Fairness and forward-progress

- What about Opacity?

- Interaction with PTW, exceptions, ...
Wrapping Up

- ARM’s new, stronger v8.0 memory model
  - Multi-copy atomicity
  - Store token optimisation (disallowed)
  - False data and control dependencies (enforced)

https://github.com/herd/herdtools7/commit/daa126680b6ecba97ba47b3e05bbaa51a89f27b7

- Extending weak memory with transactional memory

- We are hiring! Verification and specification of real-world systems
  http://www.arm.com/careers (search 10720)
References

[AMT14] Herding cats: Modelling, Simulation, Testing, and Data-mining for Weak Memory (Alglave, Maranget, Tautschnig)
[CMF+13] Robust Architectural Support for Transactional Memory in the Power Architecture (Cain et al.)
[DS09] Strong Isolation is a Weak Idea (Dalessandro, Scott)
[FSP+17] Mixed-size concurrency: ARM, POWER, C/C++11, and SC (Flur et al.)
[HM93] Transactional Memory: Architectural Support for Lock-Free Data Structures (Herlihy, Moss)
[MBL06] Subtleties of Transactional Memory Atomicity Semantics (Martin, Blundell, Lewis)
[WBS+17] Automatically Comparing Memory Consistency Models (Wickerson, Batty, Sorensen, Constantinides)