Mapping C/C++ Concurrency to Processors
Recap: Transformations

Redundant access elimination such as
- \( Y_{rlx} = v; Y_{rlx} = v' \leadsto Y_{rlx} = v' \)  
- \( Y_{rlx} = v; a = Y_{rlx} \leadsto Y_{rlx} = v; a = v \)  
- \( a = Y_{rlx}; a = Y_{rlx} \leadsto a = Y_{rlx} \)  

(OW)

(RAW)

(RAR)

Fence Insertion such as
- \( C_1; C_2 \leadsto C_1; F_{rel,acq,sc}; C_2 \)

Access strengthening such as
- \( Y_{rlx} = v \leadsto Y_{rel} = v \)
- \( a = Y_{rlx} \leadsto a = Y_{acq} \)

Reordering of accesses such as
- \( a = Y_{rlx}; Z_{rlx} = v \leadsto Z_{rlx} = v; a = Y_{rlx} \)
GCC and LLVM compiles C11 concurrency primitives.

Introduce primitives in intermediate representation (IR).

GCC follows C11 concurrency model but LLVM differs.

No optimization on atomic accesses.
C11 concurrency primitives = operations + memory order

Operations
- Load
- Store
- ...

Memory orders:
- Relaxed
- Release
- Acquire
- ...

Remember the concurrency primitives in architectures:
- Load
- Store
- Fence
- ...

Mapping to Processors
<table>
<thead>
<tr>
<th>C11</th>
<th>x86</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{na}$</td>
<td>mov(R)</td>
</tr>
<tr>
<td>$R_{rlx}$</td>
<td>mov(R)</td>
</tr>
<tr>
<td>$R_{acq}$</td>
<td>mov(R)</td>
</tr>
<tr>
<td>$R_{sc}$</td>
<td>mov(R)</td>
</tr>
<tr>
<td>$W_{na}$</td>
<td>mov(W)</td>
</tr>
<tr>
<td>$W_{rlx}$</td>
<td>mov(W)</td>
</tr>
<tr>
<td>$W_{rel}$</td>
<td>mov(W)</td>
</tr>
<tr>
<td>$W_{sc}$</td>
<td>mov(W); mfence</td>
</tr>
<tr>
<td>$U_{o}$</td>
<td>lock cmpxchg</td>
</tr>
<tr>
<td>$F_{\sqsupset sc}$</td>
<td>skip</td>
</tr>
<tr>
<td>$F_{sc}$</td>
<td>mfence</td>
</tr>
</tbody>
</table>
Example

C11 to x86: Why $W_{sc} \sim W$; mfence is required.

C11: $a = b = 0$ is disallowed.

x86: $a = b = 0$ is disallowed
(but allowed without the mfence accesses)

$$X_{sc} := 1; \quad Y_{sc} := 1; \quad X := 1; \quad Y := 1;$$
$$a := Y_{sc}; \quad a := X_{sc}; \quad a := Y; \quad a := X;$$
$$\sim$$

$$\text{mfence} \quad \text{mfence}$$
Example: Alternative Scheme

C11 to x86: Why $W_{sc} \leadsto U$

C11: $a = b = 0$ is disallowed.

x86: $a = b = 0$ is disallowed
(no mfence is required)

\[
\begin{align*}
X_{sc} &:= 1; & Y_{sc} &:= 1; & U(X, 0, 1); & U(Y, 0, 1); \\
a &:= Y_{sc}; & a &:= X_{sc}; & a &:= Y; & a &:= X;
\end{align*}
\]
We prove the mapping scheme is correct.

For each consistent execution of the target program there exists a corresponding consistent execution of the source program with same outcome.

\[ X \in [P_{tgt}]. \exists X' \in [P_{src}]. O(X) = O(X') \]

1. For each consistent execution \( X' \) of target program \( P_{tgt} \), define an execution \( X \) of \( P_{src} \).
2. Show that \( X \) is consistent.
3. Show that \( O(X) = O(X') \)
For each consistent execution $X'$ of target program $P_{tgt}$, define an execution $X$ of $P_{src}$.

$P_{tgt}$ is x86 program and $P_{src}$ is C11 program.

$X'$ is an x86 consistent execution of target program $P_{tgt}$ and $X$ is the C11 consistent execution of source program $P_{src}$.

Let's start with non-SC programs.

In that case target execution $X'$ has no mfence.
Mapping Correctness

Given $X'$ we define $X$.

Based on $\mathbb{P}_{\text{src}}$:

$X'.R \implies X.R_{\text{na}}/X.R_{\text{rlx}}/X.R_{\text{acq}}$

$X'.W \implies X.W_{\text{na}}/X.W_{\text{rlx}}/X.W_{\text{rel}}$

$X'.U \implies X.U_{\text{na}}/X.U_{\text{rlx}}/X.U_{\text{rel}}/X.U_{\text{acq}}/X.U_{\text{acq\_rel}}$

It is sufficient to consider:

$X'.R \implies X.R_{\text{acq}}$

$X'.W \implies X.W_{\text{rel}}$

$X'.U \implies X.U_{\text{acq\_rel}}$

Question: Why is it sufficient?
C11 Source Execution

\[ a \in X'.E \iff a \in X.E \text{ where} \]
- \[ a' \in R \iff a \in R_{\text{acq}} \]
- \[ a' \in W \iff a \in W_{\text{rel}} \]
- \[ a' \in U \iff a \in U_{\text{acq}_\text{rel}} \]

\[ X'.rf(a', b') \iff X.rf(a, b) \]

\[ X'.so(a', b') \iff X.mo(a, b) \]

Relations in C11:
\[(X.po \cup X.rf)^+ = (X.po \cup X.sw)^+ = X.hb\]
We show $X$ is consistent.

(irrHB)
Assume a $X.hb$ cycle.
It implies a $(X.po \cup X.rf)^+ cycle.
It implies a $(X'.po \cup X'.rf)^+$ cycle which is a contradiction.
Therefore $X$ satisfies (irrHB).
We show $X$ is consistent.

(irrHB)
Assume a $X.hb$ cycle.
It implies a $(X.po \cup X.rf)^+$ cycle.
It implies a $(X'.po \cup X'.rf)^+$ cycle which is a contradiction.
Therefore $X$ satisfies (irrHB).

(CohHBRF)
Assume a $X.rf; X.hb$ cycle.
It implies a $X.rf; (X.po \cup X.rf)^+$ cycle.
It implies a $(X'.po \cup X'.rf)^+$ cycle which is a contradiction.
Therefore $X$ satisfies (CohHBRF).
(Coh-WW)
Assume a $X.\text{mo}; X.\text{rf}; X.\text{hb}$ cycle.
It implies a $X'.\text{so}; X.\text{hb}$ cycle.
However, $X'$ satisfies (ConsHBSO) and therefore a contradiction.
Hence $X$ satisfies (Coh-WW).

(Coh-WR)
Assume a $X.\text{mo}; X.\text{hb}; X.\text{rf}^{-1}$ cycle.
It implies a $X.\text{hb}; X.\text{fr}$ cycle.
It implies a $X'.\text{hb}; X'.\text{fr}$ cycle.
However, $X'$ satisfies (ConsFRHB) and therefore a contradiction.
Hence $X$ satisfies (Coh-WR).
(Coh-RR)
Assume a $X.\text{mo}; X.\text{rf}; X.\text{hb}; X.\text{rf}^{-1}$ cycle.
It implies a $X.\text{mo}; X.\text{hb}; X.\text{rf}^{-1}$ cycle.
It implies a $X.\text{hb}; X.\text{fr}$ cycle.
It implies a $X'.\text{hb}; X'.\text{fr}$ cycle.
However, $X'$ satisfies (ConsFRHB) and therefore a contradiction.
Hence $X$ satisfies (Coh-RR).

(Atomicity)
Assume a $X.\text{fr}; X.\text{mo}$ cycle.
It implies a $X'.\text{fr}; X'.\text{so}$ cycle.
However, $X'$ satisfies (ConsFRSO) and therefore a contradiction.
Hence $X$ satisfies (Atomicity).