

INIT: A=2; B=0; X=0; Lb=Lx=0

T1

T2

T3

lock(Lx)
X:=X+2;
unlock(Lx)

lock(Lb)
B:=B+1
unlock(Lb)

while(B<3);
<wait>

C:=D*E
C:=C+K
.....

lock(Lx)
X:=A+X
unlock(Lx)

lock(Lb)
B:=B+1
unlock(Lb)

while(B<3);
<wait>

Y:=2*Y
Z:=D+2
.....

lock(Lb)
B:=B+1
unlock(Lb)

while(B<3);
<wait>

Locks use special instructions

- RMW (for lock)
- special stores (for unlocks)

Treat locks as FENCES for all memory accesses

Also called "SYNC" in PowerPC's memory model

lock and unlock act as fences

- perform all preceeding accesses in t.o. before attempting a SYNC
- perform SYNC before attempting any following accesses in t.o.

In between two SYNC Ops, order of memory Ops is arbitrary