1)

Test and set:

While(locked.exchange(true)) {

//pass

}

TATAS:

While(locked.exchange(true)) {

While(locked.load()) {}

}

1. Когда N потоков накидываются на TAS спинлок и один из них его захватывает, остальные начинают крутиться в цикле, читать и перезаписывать значение locked. Чтобы прочитать значение locked, кэш-линия должна стать shared, чтобы перезаписать значение locked, - modified, кэш линиям других потоков посылается сигнал на инвалидацию кэш-линий (в шине сообщения read invalidate – в ответ – read response, invalidate aknowledge), значит, вызов exchange в одном потоке делает M кэш-линию этого потока, все остальные инвалидируются. Вызов в другом потоке exchange – произойдет выгрузка в память значения в modified, modified и invalid станут shared, затем этот поток снова захочет изменить значение, станет modified… Большое количество сообщений в кэш-линии, много выгрузок в память и изменения состояний кэш-линий.

В случае TATAS спинлок – один поток заходит в критическую секцию, остальные будут крутиться в цикле и читать значение locked – все находятся в состоянии shared, нет выгрузок в память и сообщений, поэтому TATAS спинлок лучше

1. Потоки, которые крутятся в состоянии load – в состоянии shared, когда поток из критической секции изменяет значение, он делает свою кэш-линию modified, остальным отправляется сообщение на инвалидацию, потоки снова делают locked.load() – читают значение, значит, выгрузка в память значения из modified, все кэш-линии(и потока в критической секции и в цикле) становятся shared.