SoundWire Locking

This document explains locking mechanism of the SoundWire Bus. Bus uses following locks in order to avoid race conditions in Bus operations on shared resources.

- Bus lock
- Message lock

Bus lock

SoundWire Bus lock is a mutex and is part of Bus data structure (sdw_bus) which is used for every Bus instance. This lock is used to serialize each of the following operations(s) within SoundWire Bus instance.

- Addition and removal of Slave(s), changing Slave status.
- Prepare, Enable, Disable and De-prepare stream operations.
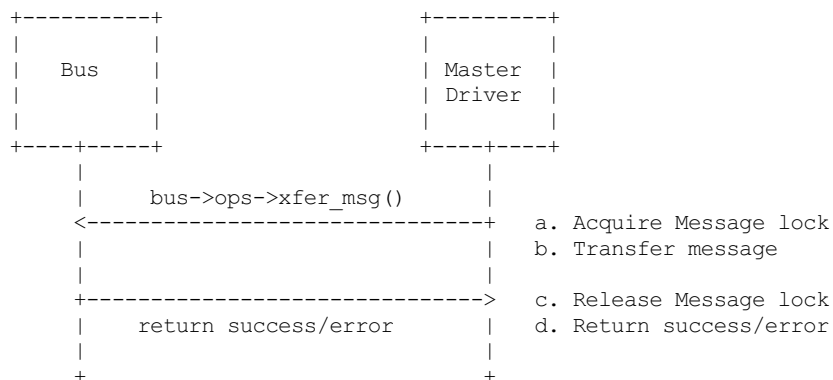- Access of Stream data structure.

Message lock

SoundWire message transfer lock. This mutex is part of Bus data structure (sdw_bus). This lock is used to serialize the message transfers (read/write) within a SoundWire Bus instance.

Below examples show how locks are acquired.

Example 1

Message transfer.

1. For every message transfer

   a. Acquire Message lock.
   b. Transfer message (Read/Write) to Slave1 or broadcast message on Bus in case of bank switch.
   c. Release Message lock

```
+---------+                    +---------+
|         |                    |         |
|  Bus    |                    | Master  |
|         |                    | Driver  |
|         |                    |         |
+----+----+                    +----+----+
     |                              |
     |      bus->ops->xfer_msg()    |
     <------------------------------+   a. Acquire Message lock
     |                              |   b. Transfer message
     |                              |
     +------------------------------>   c. Release Message lock
     |    return success/error      |   d. Return success/error
     |                              |
     +                              +
```

Example 2

Prepare operation.

1. Acquire lock for Bus instance associated with Master 1.

2. For every message transfer in Prepare operation

   a. Acquire Message lock.
   b. Transfer message (Read/Write) to Slave1 or broadcast message on Bus in case of bank switch.
   c. Release Message lock.

3. Release lock for Bus instance associated with Master 1

```
+---------+                    +---------+
|         |                    |         |
|  Bus    |                    | Master  |
|         |                    | Driver  |
|         |                    |         |
+----+----+                    +----+----+
     |                              |
     |     sdw_prepare_stream()     |
     <------------------------------+   1. Acquire bus lock
```

```
|                             |  2. Perform stream prepare
|                             |
|                             |
|      bus->ops->xfer_msg()   |
<-----------------------------+  a. Acquire Message lock
|                             |  b. Transfer message
|                             |
+----------------------------->  c. Release Message lock
|     return success/error    |  d. Return success/error
|                             |
|                             |
|     return success/error    |  3. Release bus lock
+----------------------------->  4. Return success/error
|                             |
+                             +
```