TẬP ĐOÀN CÔNG NGHIỆP
VIỄN THÔNG-QUÂN ĐỘI
VIETTEL

DESIGN SPECIFICATION

MODEL

BUS
MEMORY MAPPING I/O

Author: Huan Nguyen-Duy
Date: 7/19/2024

CATALOG

FIGURE

TABLE

Error! No table of figures entries found.

1. OVERVIEW

The document illustrates the design specification for BUS architecture using memory mapping I/O (MMIO) with clock synchronization. The BUS MMIO contains multiple target and initiator sockets that are used to connect with multiple masters and slaves. In particular, each slave registers a specific address space and identification.

2. BUS ARCHITECTURE

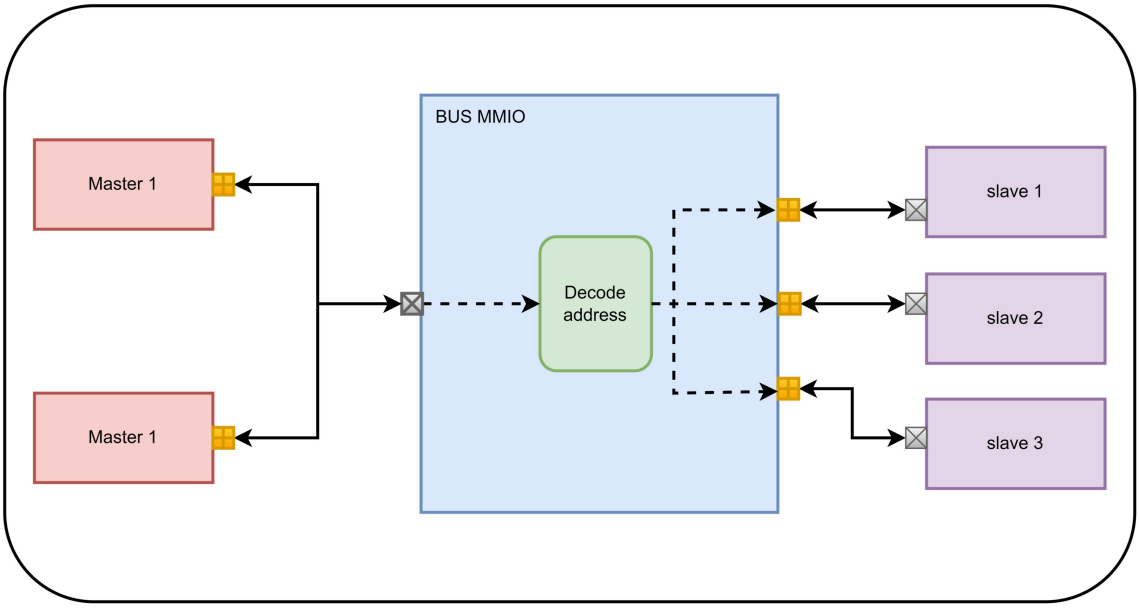

Figure 1: BUS architecture

The bus architecture is illustrated in Fig 1. In particular, bus MMIO can connect multiple masters and slaves, addressing the destination of transaction by decode address block, it decodes the address of transaction that is sent by master and forward forwards it to the specific slave.

Many masters model can be bind to bus MMIO through a simple target socket provided by TLM 2.0 library.

The user can define the number of initiator socket for the bus MMIO, each slave can be mapped into bus with specific address space and socket identification.

3. SEQUENCE DIAGRAM

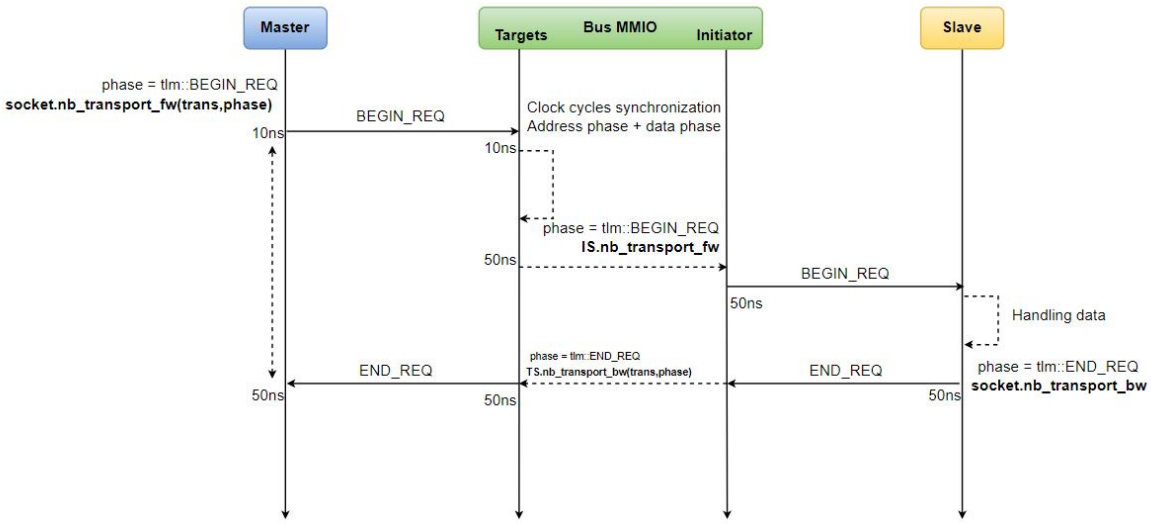

Figure 2: The sequence diagram using TLM non-blocking

The sequence diagram illustrates for the operation between a master and salve through bus MMIO. In particular, the bus MMIOP operation is implemented base on TLM AT non-blocking that is simple to 2 phase with BEGIN_REQ and END_REQ phases respectively.

At the time when the master conduces transaction, the bus MMIO decodes address and synchronizes with clock cycles base on the data length. The transaction is success when END_REQ phase that is forward from slave to master.

Depending on the type of transaction is read or write request. The write request data will be sent to slave at BEGIN_REQ phase that is forward from initiator socket with identification. On the other hand, when the read request transaction is conduces, the master will received data at the END_REQ phase.

4. FLOW DIAGRAM

In this section, the document provides flow diagrams of functions that illustrates clearly the operation of the bus MMIO.

4.1. Nb_transport_fw flow diagram

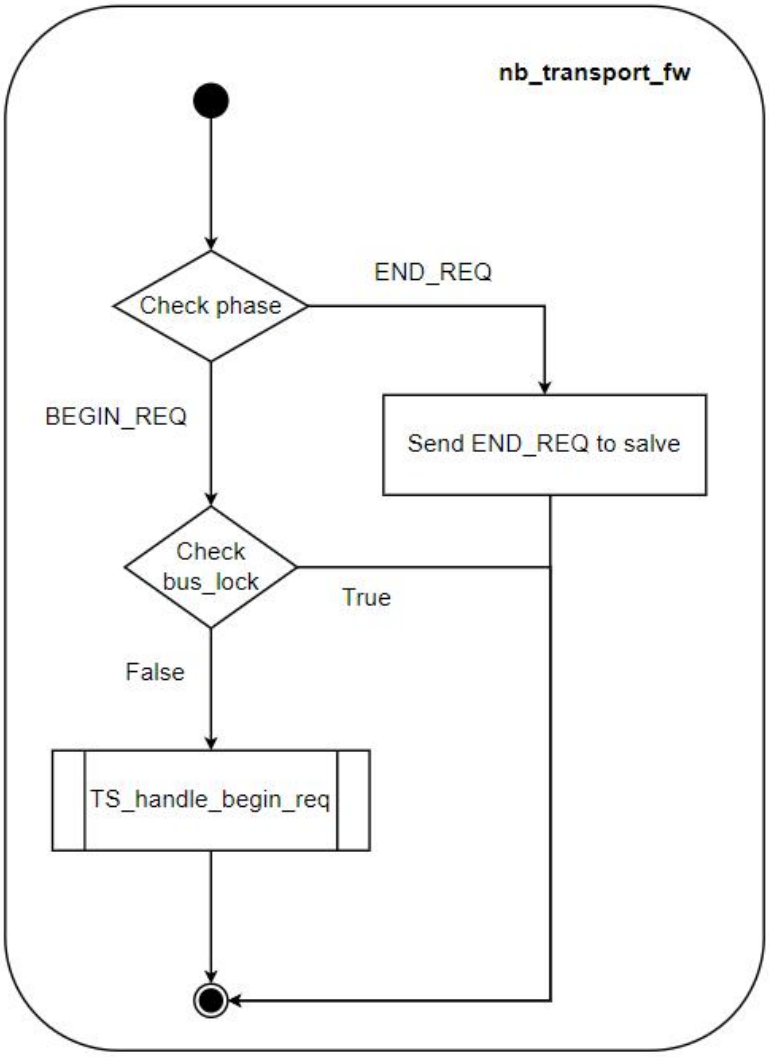

Figure 3: Nb_transport_fw flow diagram

The fig 3 illustrates for the flow diagram of call back nb_transport_fw function, it is called by the initiator socket of master and recalled by the target of the bus MMIO as call back function, the tlm generic payload and delay timing are passed through call back function nb_transport_fw.

4.2. Nb_transport_bw flow diagram

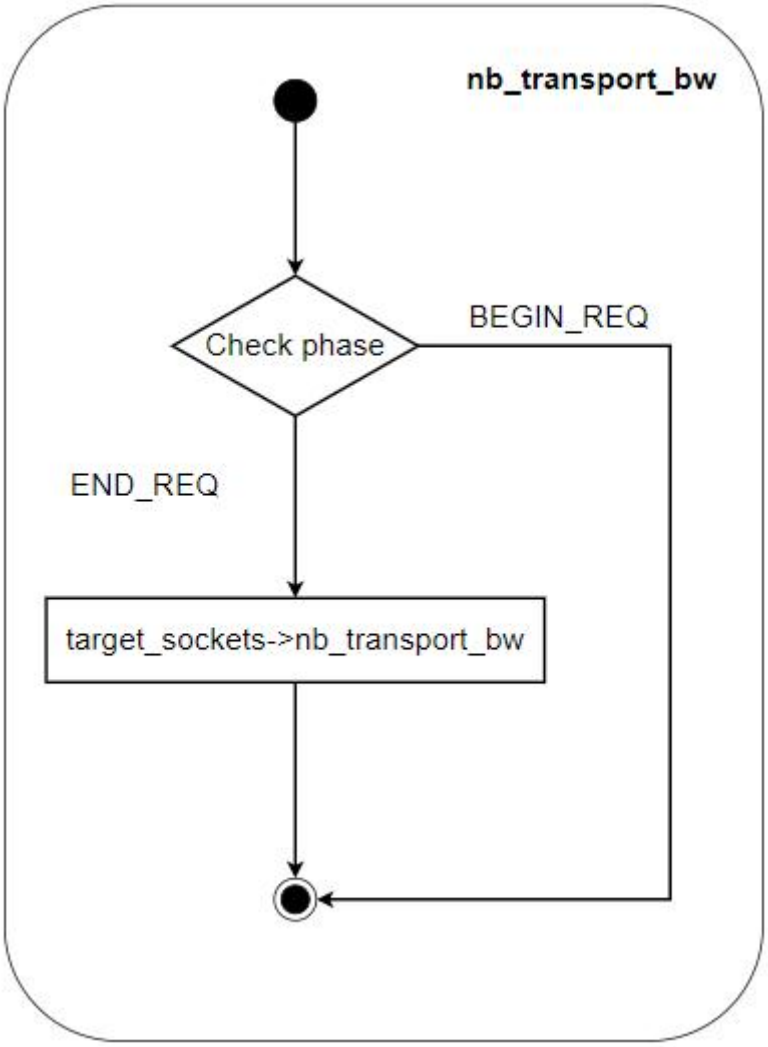

Figure 4: Nb_transport_bw flow diagram

The fig 3 illustrates for the flow diagram of call back nb_transport_bw function, it is called by the target socket of slave and recalled by the initiator socket of bus MMIO as call back function. Moreover, the tlm generic payload and timing delay are passed through this function, including its returned data.

4.3. TS_handle_begin_req flow diagram

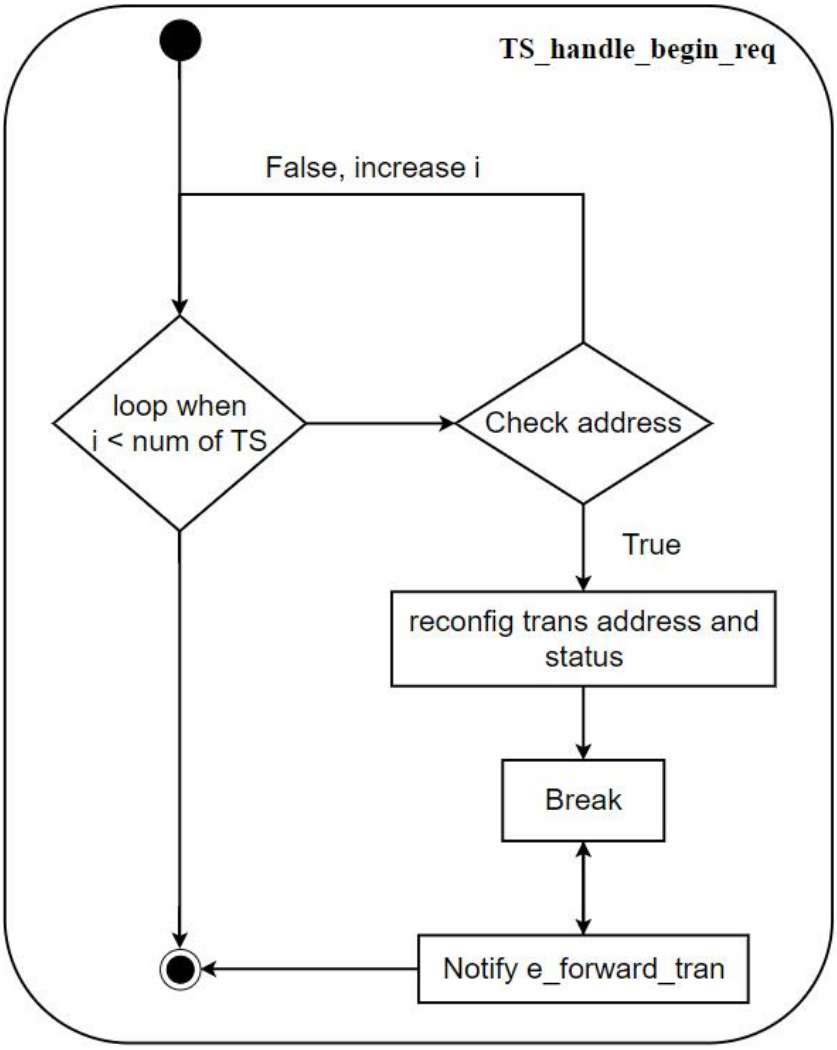

Figure 5: Ts_handle_begin_req flow diagram

The fig 5 shows the operation of Ts_handle_begin_req function, it checks the address of tlm generic payload base on the address range of target sockets that is bound into the bus MMIO in advance. If the address exists, the forward transaction event will be triggered.

4.4. Forward_transaction_process flow diagram

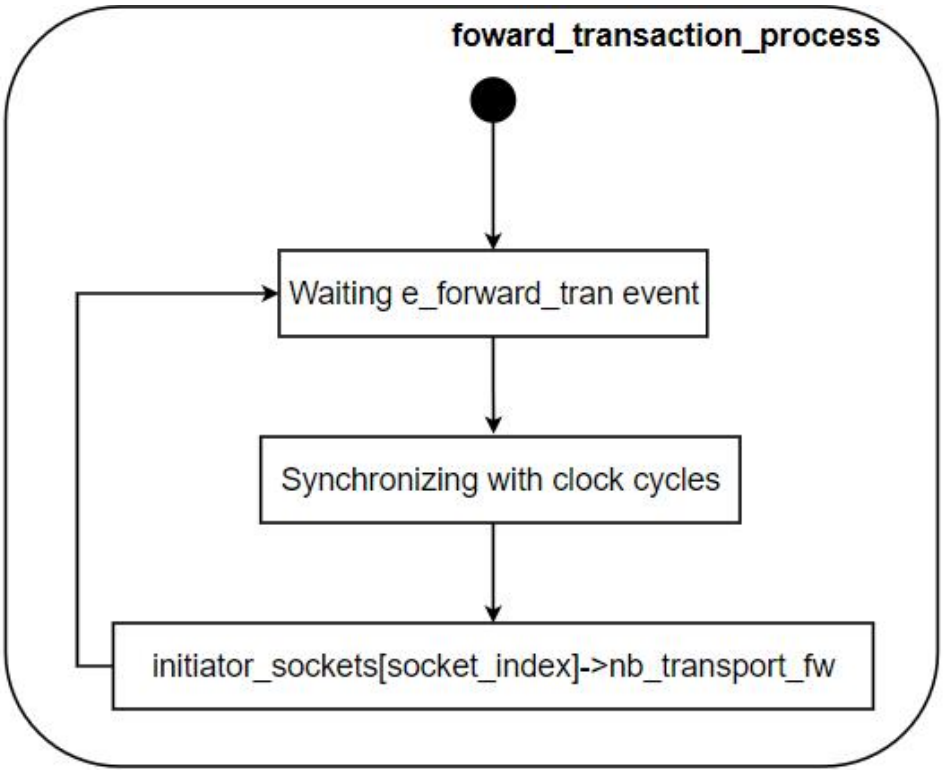

Figure 6: forward_transaction_process flow diagram

The fig 6 illustrates the operation of forward_transaction_process thread, it is triggered by e_forward_tran event, synchronizing with the number of clock cycles base on the length of transaction data. After that, the corresponding initiator socket call to nb_transport_fw to sent transaction to the slave.