On-Chip Interconnect Models

COE838/EE8221: Systems-on-Chip Design
http://www.ecb.torontomu.ca/~courses/coe838/

Dr. Gul N. Khan
http://www.ecb.torontomu.ca/~gnkhan
Electrical, Computer & Biomedical Engineering
Toronto Metropolitan University

Overview

- Single and Multiprocessor Bus Models
- Blocking and Non-Blocking Bus Transaction
- Bus Model Example
- NoC Communication
- Examples/Problems on Communication Time

Chapter 5: Computer System Design – System on Chip by M.J. Flynn and W. Luk

Simple Bus Model

Consider a computer system, where each and every Bus_request occupies the bus for the same time.

- Bus transaction time is constant

Bus occupancy can be approximated by a simple computation of the per-processor average.

- Bus is accessed by multi-users that results in contention.

Bus occupancy =

$$\rho = \text{Bus\_trans\_time}/(\text{Processor\_time}+\text{Bus\_trans\_time})$$

Multiprocessor Bus Model without Re-submission

Processor time is the mean time a processor needs to compute/process before making a bus request. Processor may overlap a fraction of its compute time with the bus time.

The Processor time is the net non-overlapped time in-between the bus requests.

Consider a simpler model: ***n*** processors accessing the bus

In the case of any event, $\rho \leq 1$.

For ***n*** processors using a single bus

Probability (processor doesn't access bus) = $(1 - \rho)$

Probability (when bus is busy) = $(1 - \rho)^n$

= Fraction of bus bandwidth realize = $B(\rho, n)$

Multiprocessor Bus Model with Resubmission

The fraction of bandwidth realized (times) the maximum bus bandwidth gives the achieved bus bandwidth, $B*w$.

The achieved bandwidth fraction (Bus occupancy) per CPU ($\boldsymbol{\rho_a}$) is given by:

$$\mathbf{n\rho_a} = \boldsymbol{B}(\rho, n) \text{ or } \mathbf{\rho_a} = \boldsymbol{B}(\rho, n)/n$$

A processor slows down by $\boldsymbol{\rho_a/\rho}$ due to bus congestion/contention.

Multiprocessor Bus Model with Request Resubmission

Given by an iterative pair of equations.

The achieved bandwidth fraction (Bus occupancy) per CPU ($\boldsymbol{\rho_a}$)

$$\mathbf{n\rho_a = 1 - (1 - a)^n}$$

and

$$\mathbf{a = \rho / [\rho + (\rho_a / \rho)(1 - \rho)]}$$

Where **a** is the actual offered request rate.

To find a final $\boldsymbol{\rho_a}$ initially set (a = ρ) to begin the iteration,

And convergence occurs in few iterations.

Processors with Blocking Transactions

Blocking Transactions: The processor is idle after the bus request is made and resumes computation only after the bus transaction is complete.

A Single Bus Master with Blocking Transactions.

There is no bus contention as the processor waits for the transaction to complete.

The achieved occupancy, $\boldsymbol{\rho_a}$ is the same as the offered occupancy.

Therefore, Occupancy $\boldsymbol{\rho = \rho_a}$

= bus_trans_time/(compute time+bus_trans_time)

Processors with non-Blocking Trans.

n Bus-Masters with Blocking Transactions

The offered occupancy is simply the $n\rho$,

where ρ is as given earlier.

Now contention can develop so we use our bus model to determine the achieved occupancy, ρ_a

ρ_a = (Bus_trans_time)/
(compute_time+Bus_trans_time+Contention_time)

Alternative case Buffered (non-blocking) Transactions

This is a complex case where a processor continues processing after making the request and it may make several requests before completion of initial request.

Bus Model Example

Suppose a processor has bus transactions that consist of cache line transfers. Assume that 80% of the transactions move a single line and occupy the bus for 20 cycles and 20% of the transactions move a double line (as in dirty line replacement), which takes 36 cycles. The mean bus transaction time is 23.2 cycles. Assume that a cache miss (transaction) occurs every 200 cycles.

i). Find the bus occupancy for a single processor system.

ii). Find the bus occupancy when there are four processors connected to the bus.

Bus Model Example contd.

Suppose a processor has bus transactions that consist of cache line transfers. Assume that 80% of the transactions move a single line and occupy the bus for 20 cycles and 20% of the transactions move a double line (as in dirty line replacement), which takes 36 cycles. The mean bus transaction time is 23.2 cycles. Assume that a cache miss (transaction) occurs every 200 cycles.

iii) Determine the contention time in the case of a 4-processor system.

NoC Communication

Consider 4×4 2D-torus: N =16 (Number of nodes or Routers) n =2 (Maximum travel distance) and its diameter is 4.

Dimension of the network and its maximum distance are important to determine NoC cost and performance.

NoC Links are characterized as following:

- *Cycle Time of the Link,* T_{ch} corresponds to time required to transmit between neighboring nodes.
- $1/T_{ch}$ is the bandwidth of a 1-bit wire in the NoC link.
- *Width of the Link, w* is the number of bits that can be concurrently transmitted between two NoC nodes.

Message data *L plus H* header bits (having address destination).

$\boldsymbol{T_{ch} \times (L + H)/w}$ will be the time required to transmit a message between two adjacent nodes.

NoC Communication

Wormhole Routing based Communication:

When a message packet is received at a node (or router), it is buffered only long enough to decode its header to determine the destination.

- As soon as the basic routing information is determined, the message is retransmitted to destination node.
- The amount of buffering required at intermediate nodes is significantly reduced.

The overall time of transmission between two given nodes:

$T_{\text{wormhole}} = Tch \times (d * h + L/w)$

where $h = [H/w]$

d is the # of hops between source & destination nodes.

NoC Communication Example-1

Consider a wormhole routing based communication in a 2D 4×4 NoC. Determine the average number of clock cycles needed to transmit 256-bits of data among various nodes of the NoC where the width of NoC link is 16, and the header size is 64-bits.

NoC Communication Example-2

Consider a wormhole routing (based) communication in a 2D 4×4 Torus NoC. Determine the minimum number of clock cycles required to transmit 1K-bits of data from source node (1,1) to destination node (3,4) where the width of NoC link is 32, and the header size is 32 bits.