Digital Circuits and Systems
Lecture 13 Interfaces, Interconnect, and Memory

Tian Sheuan Chang

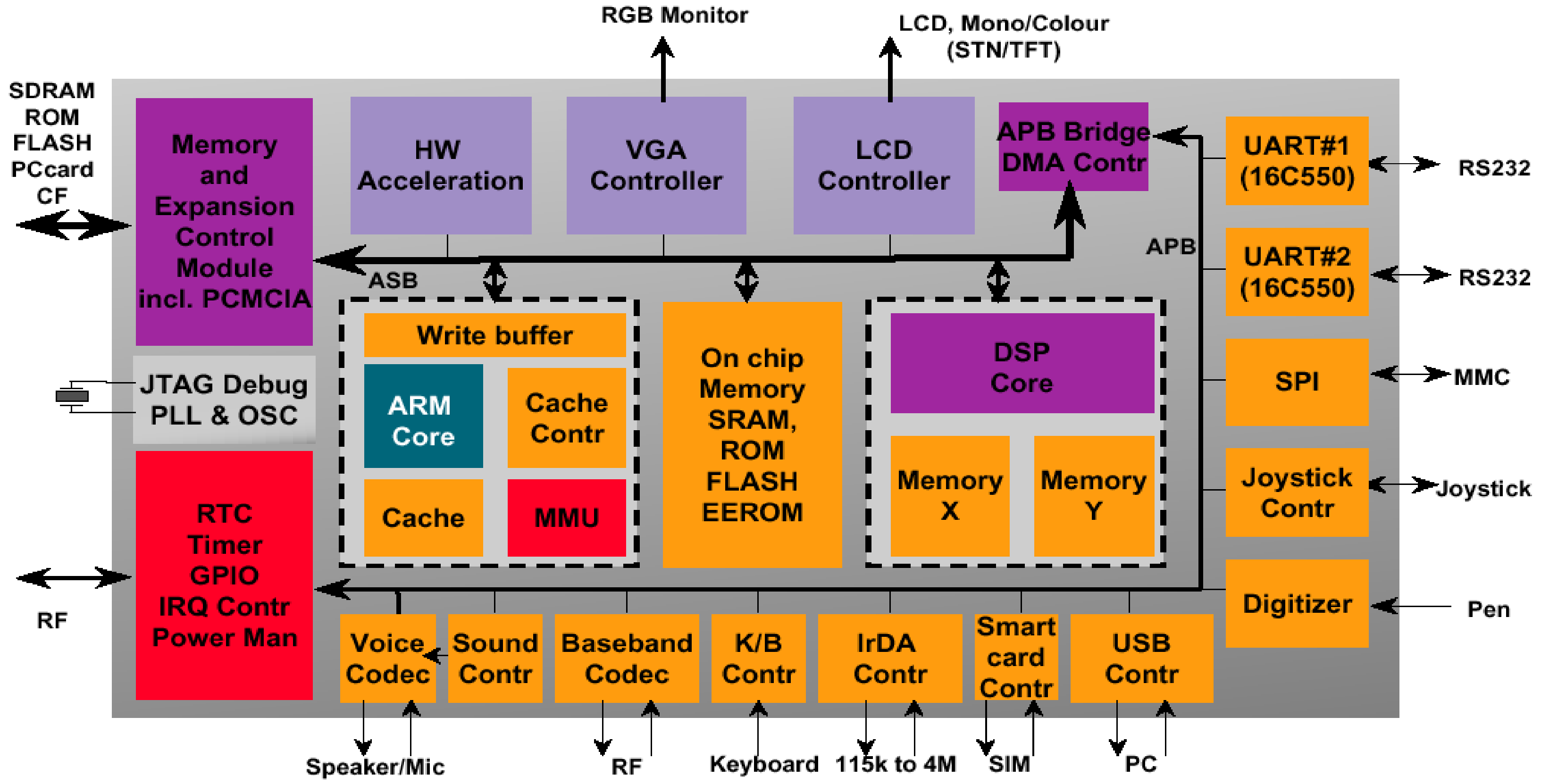
資料要從一個模組傳到另一個模組，要如何溝通，告訴對方這是有效的資料?

Outline [Dally. Ch. 22. 24. 25]

- Interface Timing: sequencing the transfer of data
  - 安排資料怎麼傳
  - 從簡單到複雜的方式
- Interconnect
  - 實體電路的樣子
  - 從簡單到複雜的方式
- Memory
  - 先存起來，慢慢再用

Recap: System on a Chip (SOC)

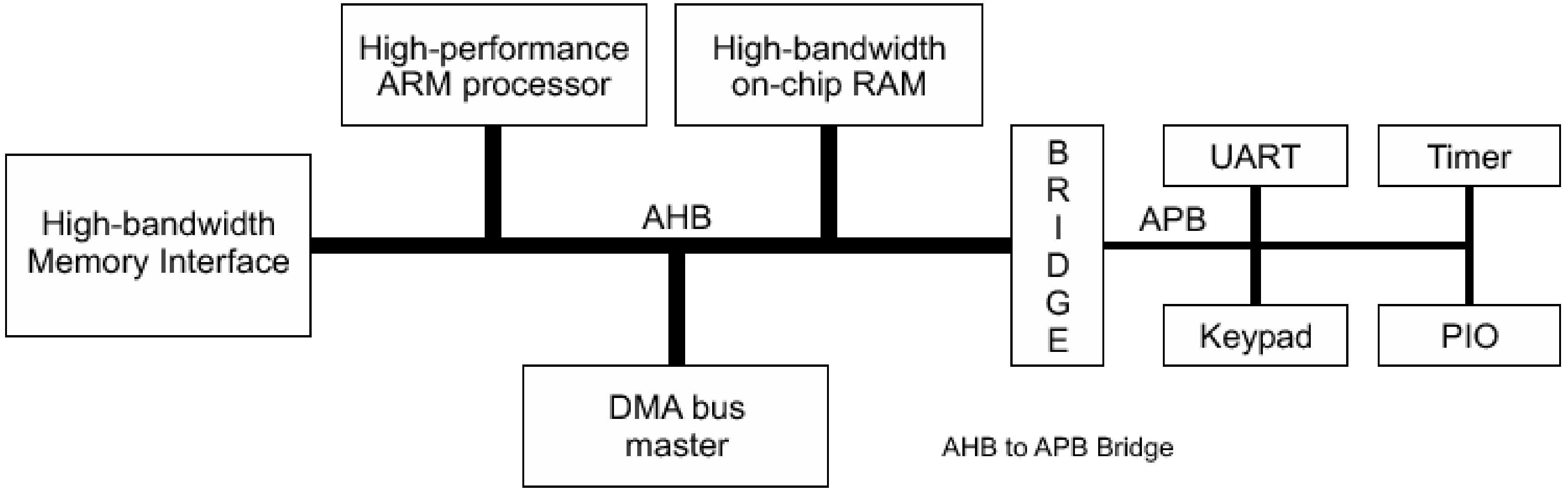
Source: ARM



Recap: Breakdown of Modern SOC Design

- Core
  - CPU/DSPs/GPUs
  - Hardware accelerators (AI engine/video codec/MP3 codec)
  - Complex data path + FSMs (pipeline/parallel)
- Bus
  - Connect core and peripherals
  - MUX/arbiters
- Peripheral
  - Interface from/to the external
  - speech/audio, camera/display, I2S/I2C, GPIO
  - Serial to parallel, parallel to serial + FSM

Recap: An Example AMBA System

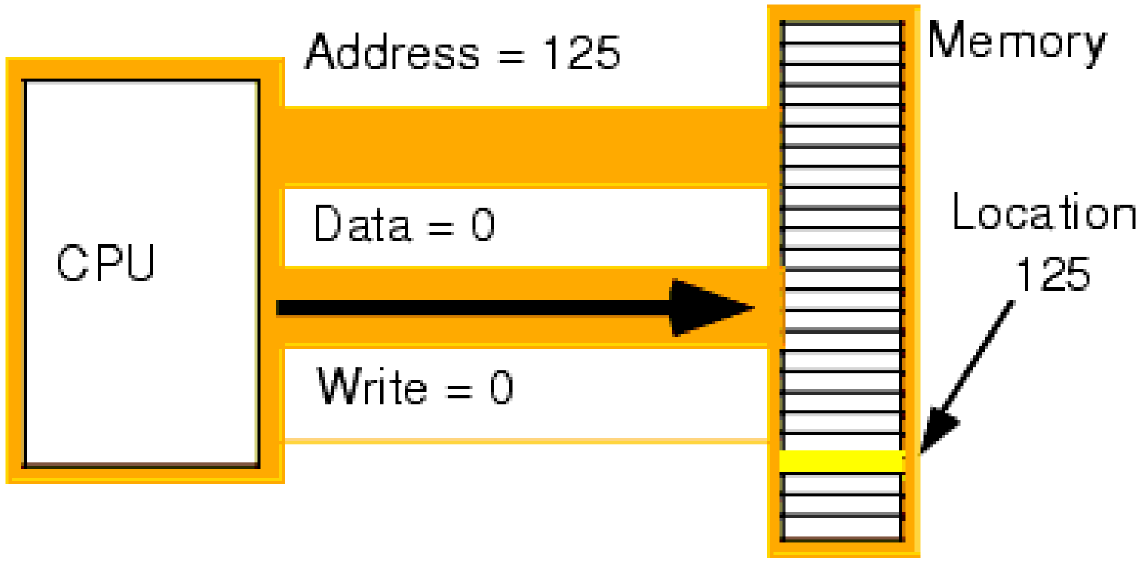

High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth Memory Interface
AHB
DMA bus master
B R I D G E
APB
UART
Timer
Keypad
PIO
AHB to APB Bridge

AMBA Advanced High-performance Bus (AHB)
* High performance
* Pipelined operation
* Burst transfers
* Multiple bus masters
* Split transactions

AMBA Advanced Peripheral Bus (APB)
* Low power
* Latched address and control
* Simple interface
* Suitable for many peripherals

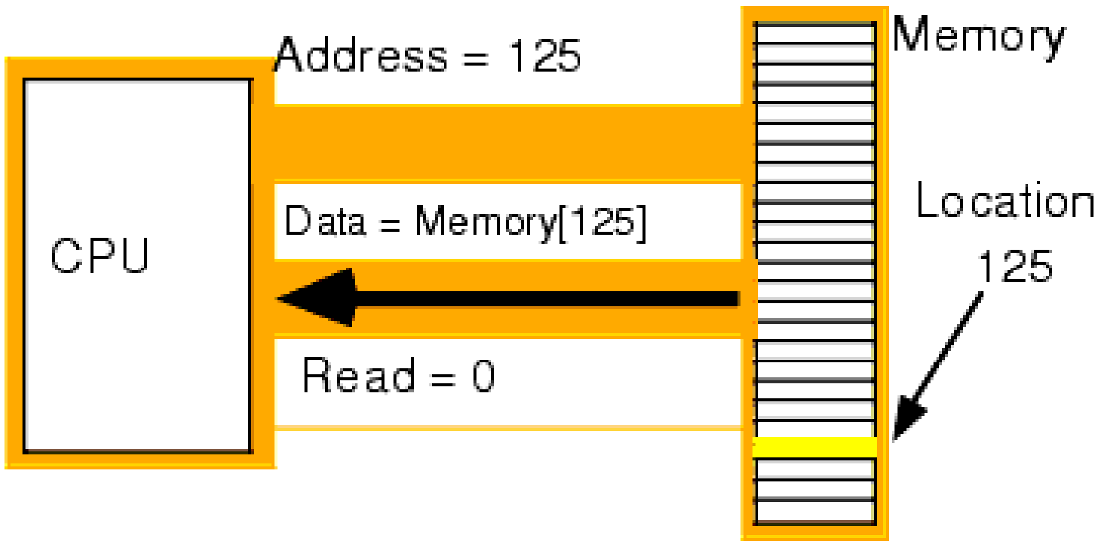
Recap: From C Array to Memory Access

Memory [125] = 0



Memory Write Operation

CPU = Memory [125]

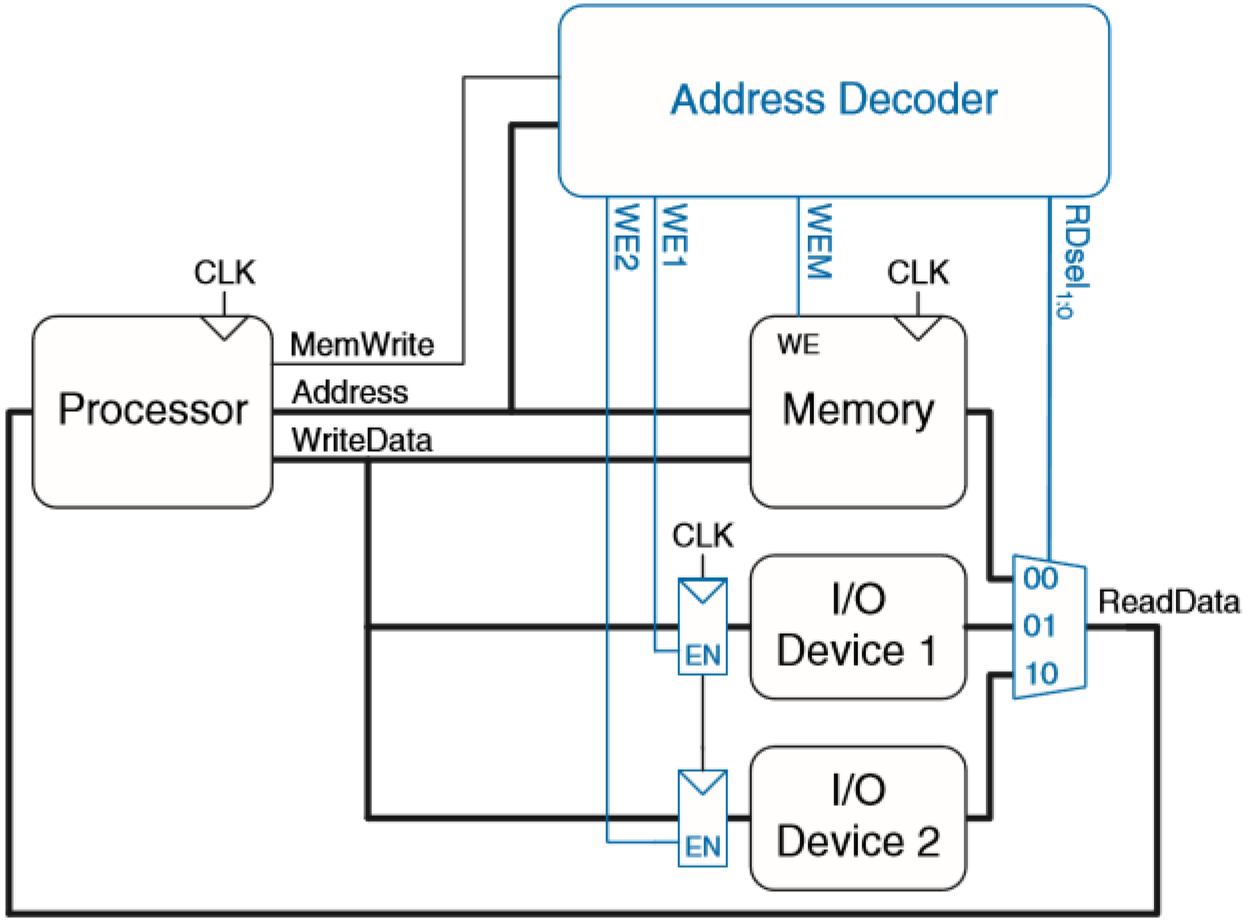


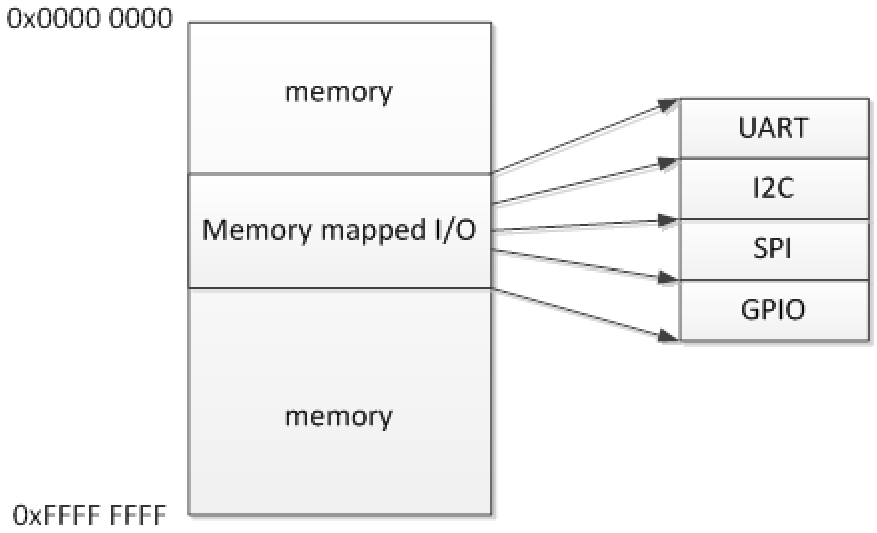
Memory Read Operation

http://www.phatcode.net/res/260/files/html/SystemOrganizationa2.html

Recap: How CPU access these I/O interface?

- Memory mapped I/O (treat other components as memory)

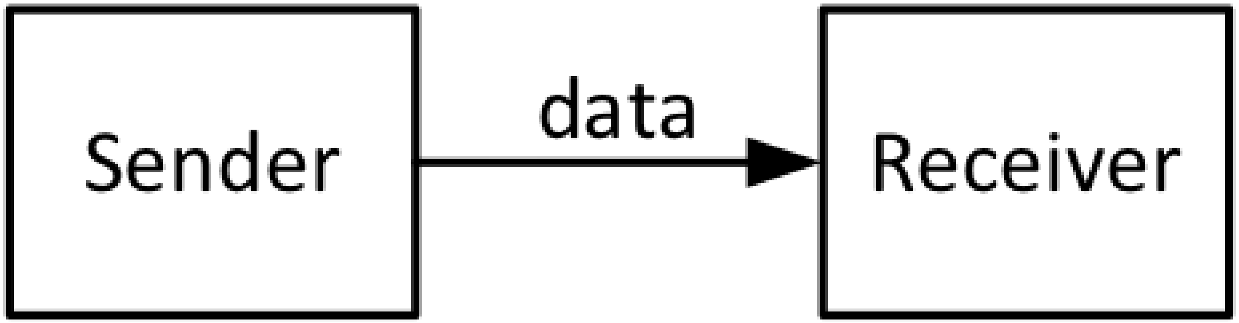




INTERFACE TIMING

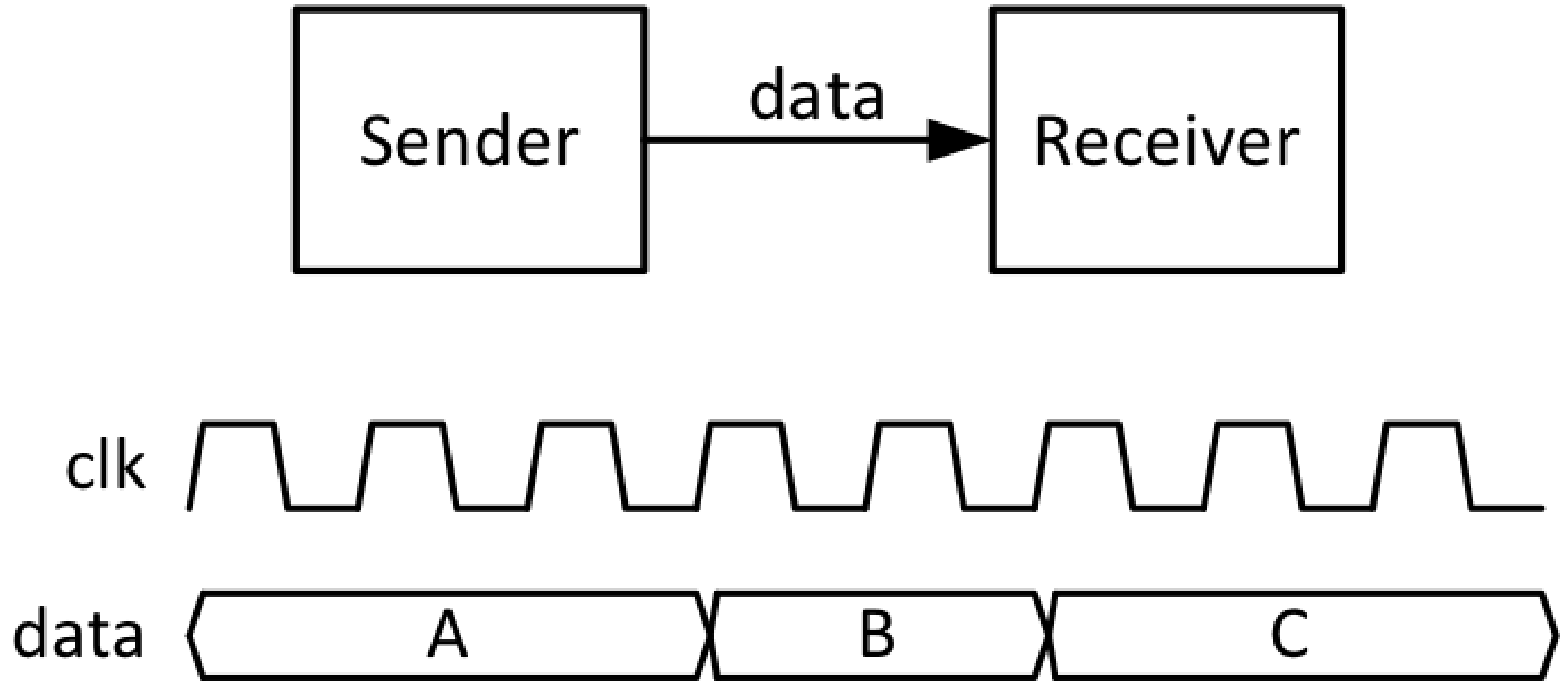
Interface Timing

- How do you pass data from one module to another?
  - Open loop: 資料送了就送了，有沒有收到是你家的事，繼續送下一筆
  - Flow control: 資料送了，要有通關訊號 或 回條，才送下一筆
  - Serialized: 通道狹窄，一次只能傳一個bit
  - 比較複雜的通常都是晶片外面溝通用，或大模組間使用，因為不確定高



(c) 2005-2012 W. J. Dally

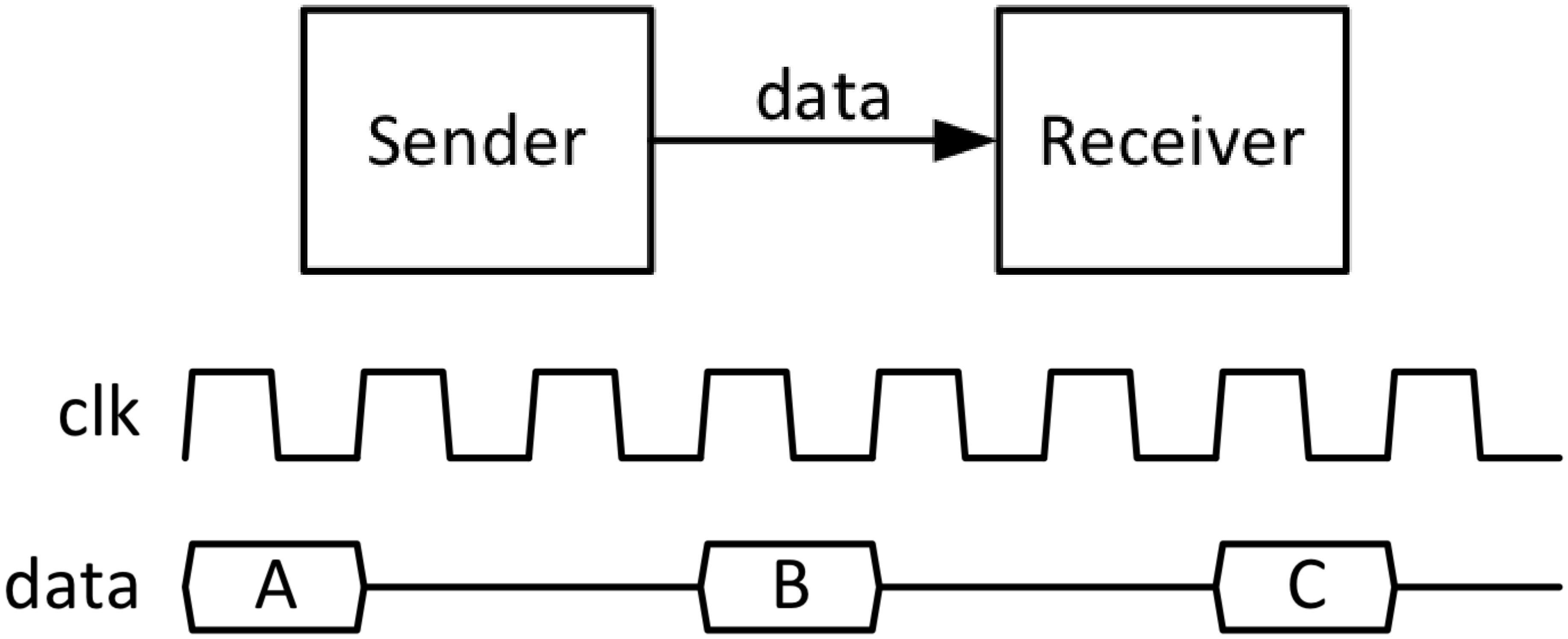
簡單的方式
Always Valid Timing



(通常都是這種，晶片內部用較多，確定性高)
Dataum is valid every cycle, and can be sampled by the receiver at any time

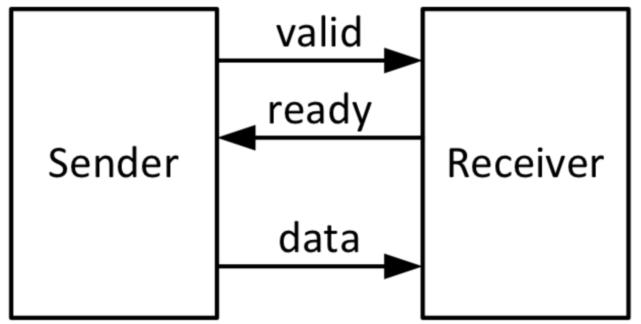
(c) 2005-2012 W. J. Dally

簡單的方式
Periodically Valid Timing

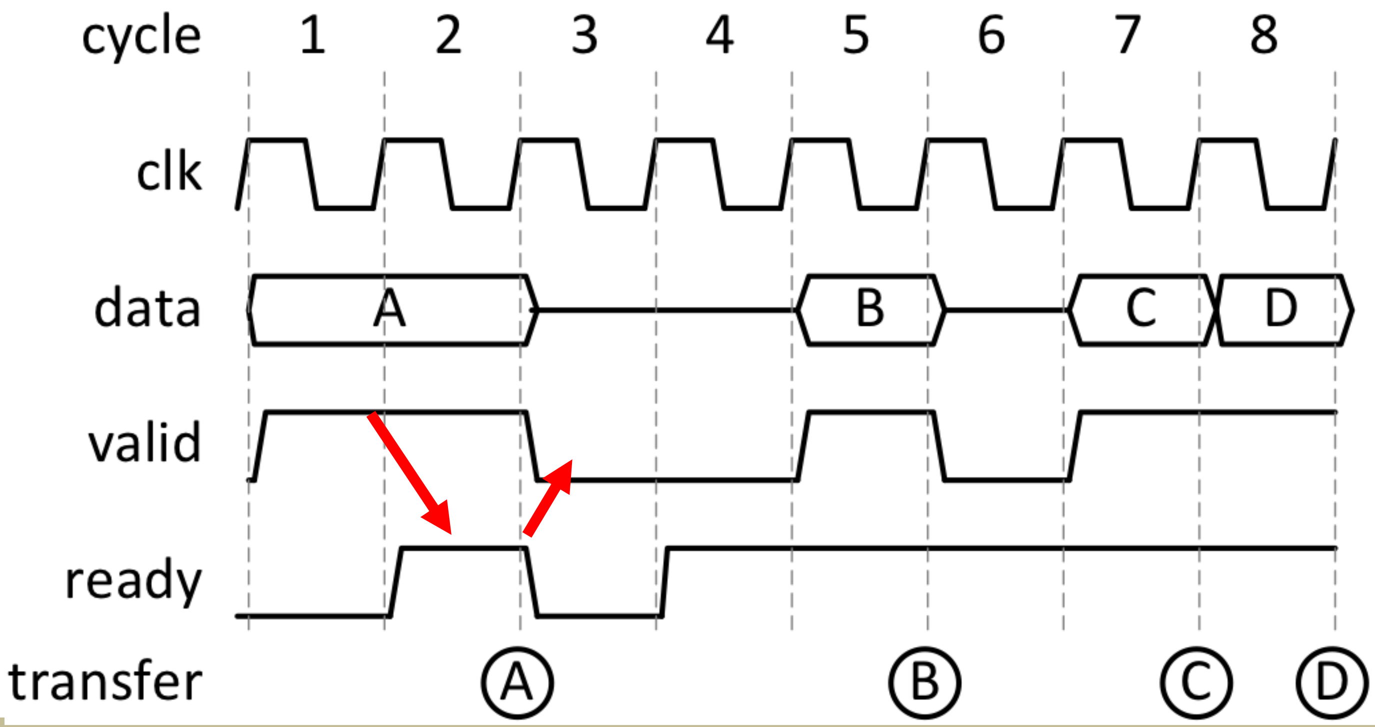


通常會有一個data_valid 訊號，表示資料在data_valid ==1 那段時間有效，可以讀，規則時就省略
不規則出現會需要明白此訊號表示
No flow control, and values cannot be dropped or repeated
Need flow control to move the valid signal across clock domain

複雜的方式
Flow Control: ready-valid flow

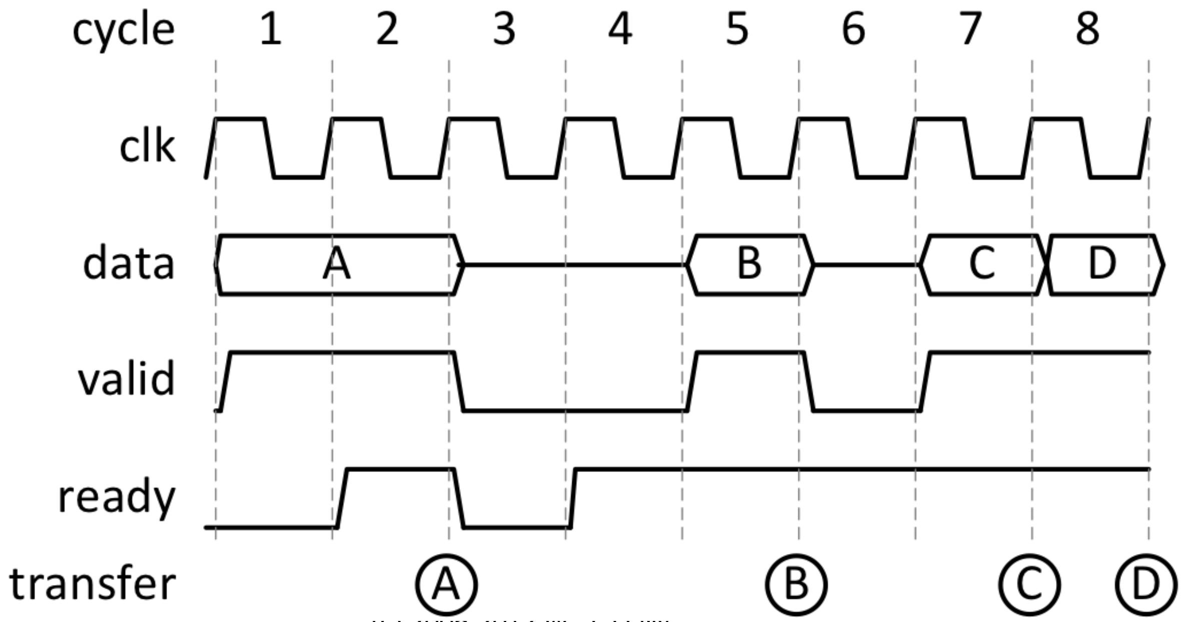


Ready 或valid 兩者都為1
才能傳

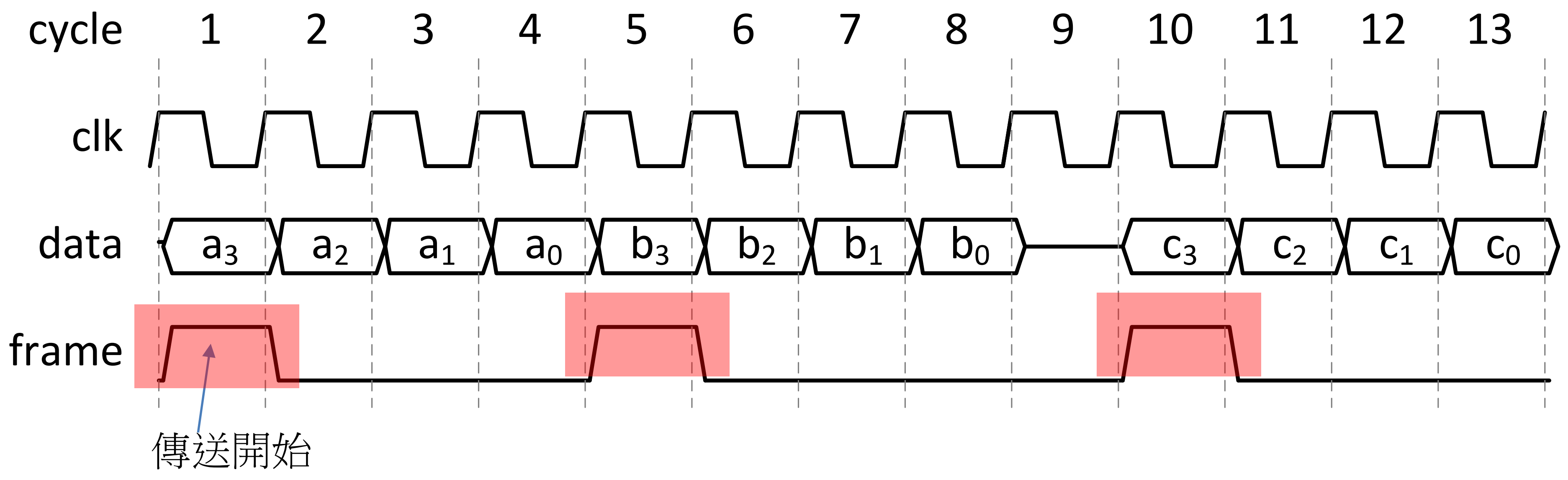


Flow-control

- Valid – Tx has data available 傳送端資料是有效的
- Ready – Rx able to take data 通關訊號，接收端可以收資料
- 特例
  - Push flow control – assume Rx always Ready
  - Pull flow control – assume Tx always Valid



複雜的方式 Serialization: USB, SATA, I^{2}C, I^{2}S

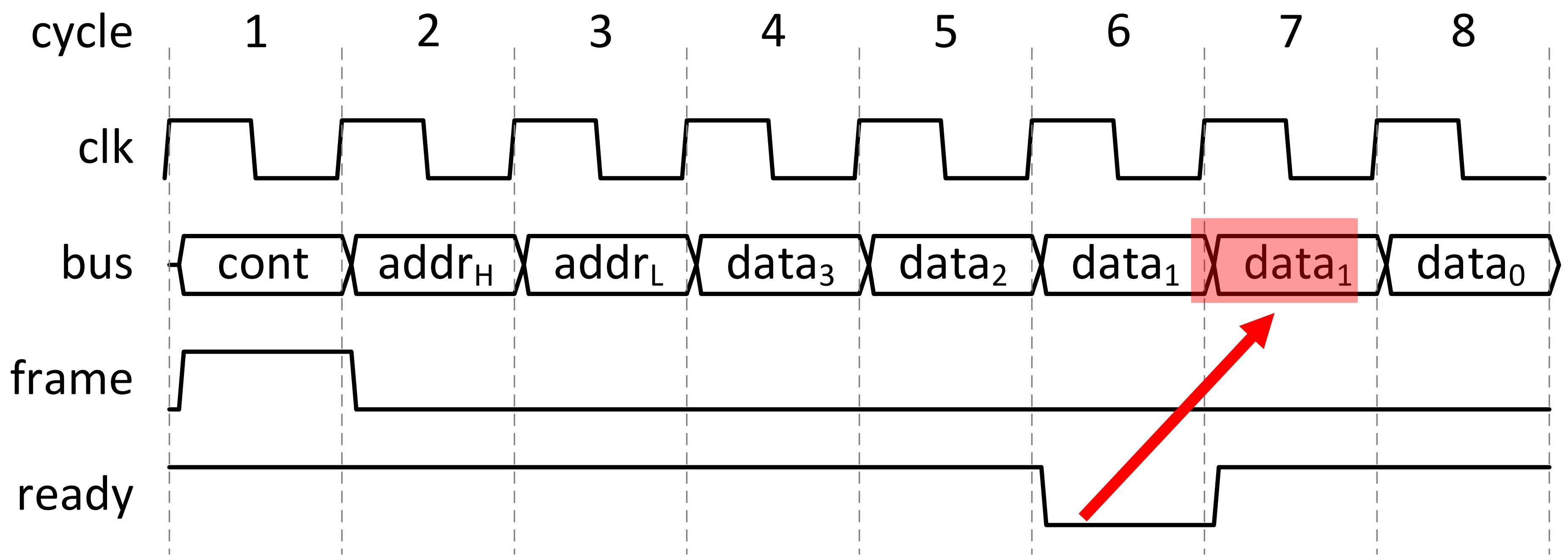


Frame signals start of new serial frame (in this case 4 words)

Flow control can be **at frame granularity** or **word granularity**

高頻或介面接腳數量有限狀況時，會用這個方法

Serialization with word granularity FC

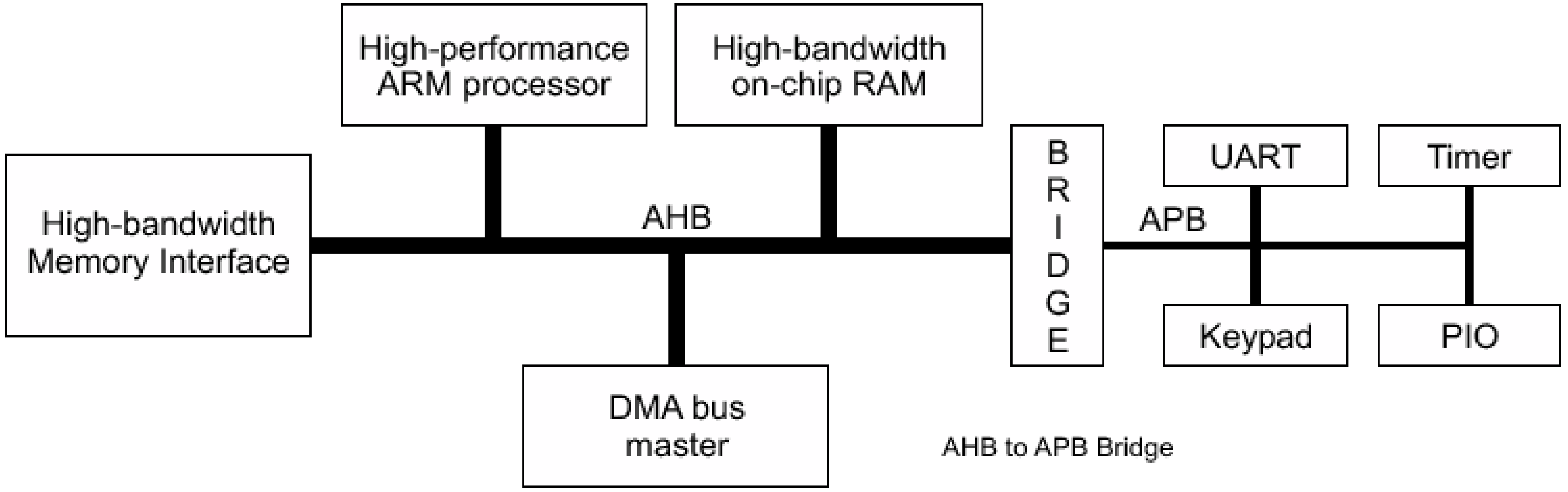


不能接收，資料要再延續下去

常見介面訊號

- Control
  - Read/write, sequential, non-sequential
- Address
  - 讀寫位置
  - 特別是用於bus or memory
- Data
  - 有時候會再細分成不同field，方便使用

An Example AMBA System

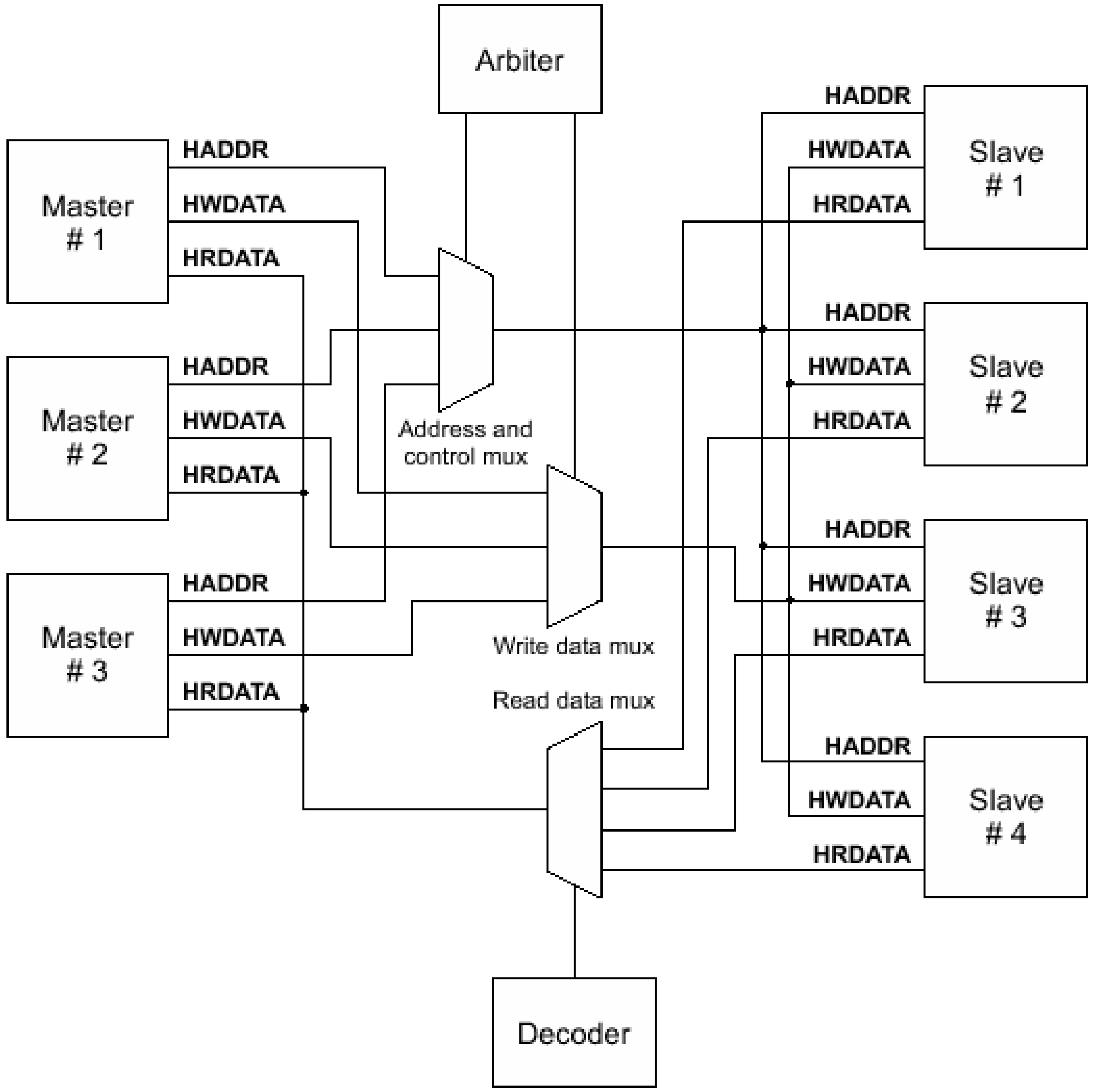


High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth Memory Interface
AHB
DMA bus master
BRIDGE
APB
UART
Timer
Keypad
PIO
AHB to APB Bridge

AMBA Advanced High-performance Bus (AHB)
* High performance
* Pipelined operation
* Burst transfers
* Multiple bus masters
* Split transactions

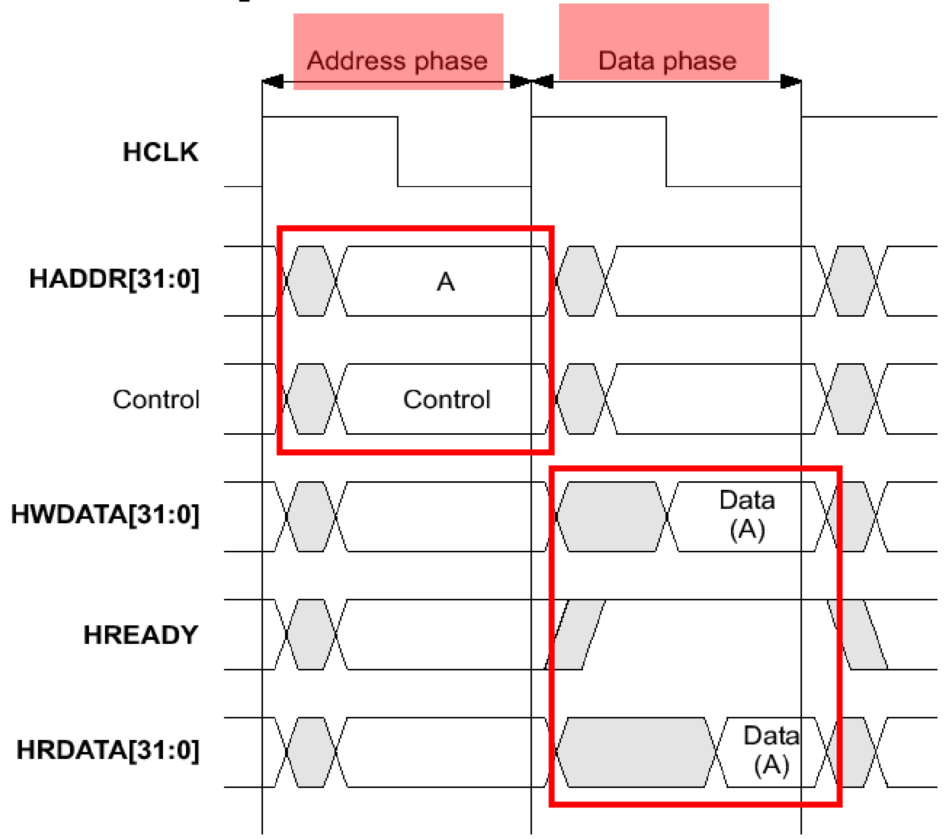
AMBA Advanced Peripheral Bus (APB)
* Low power
* Latched address and control
* Simple interface
* Suitable for many peripherals

AHB Interconnect



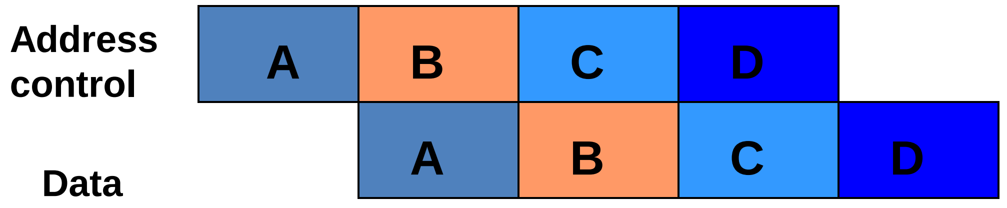
Basic components:
- Master
- Slave
- Arbiter
- Decoder
- Mux

AHB Simple Transfer

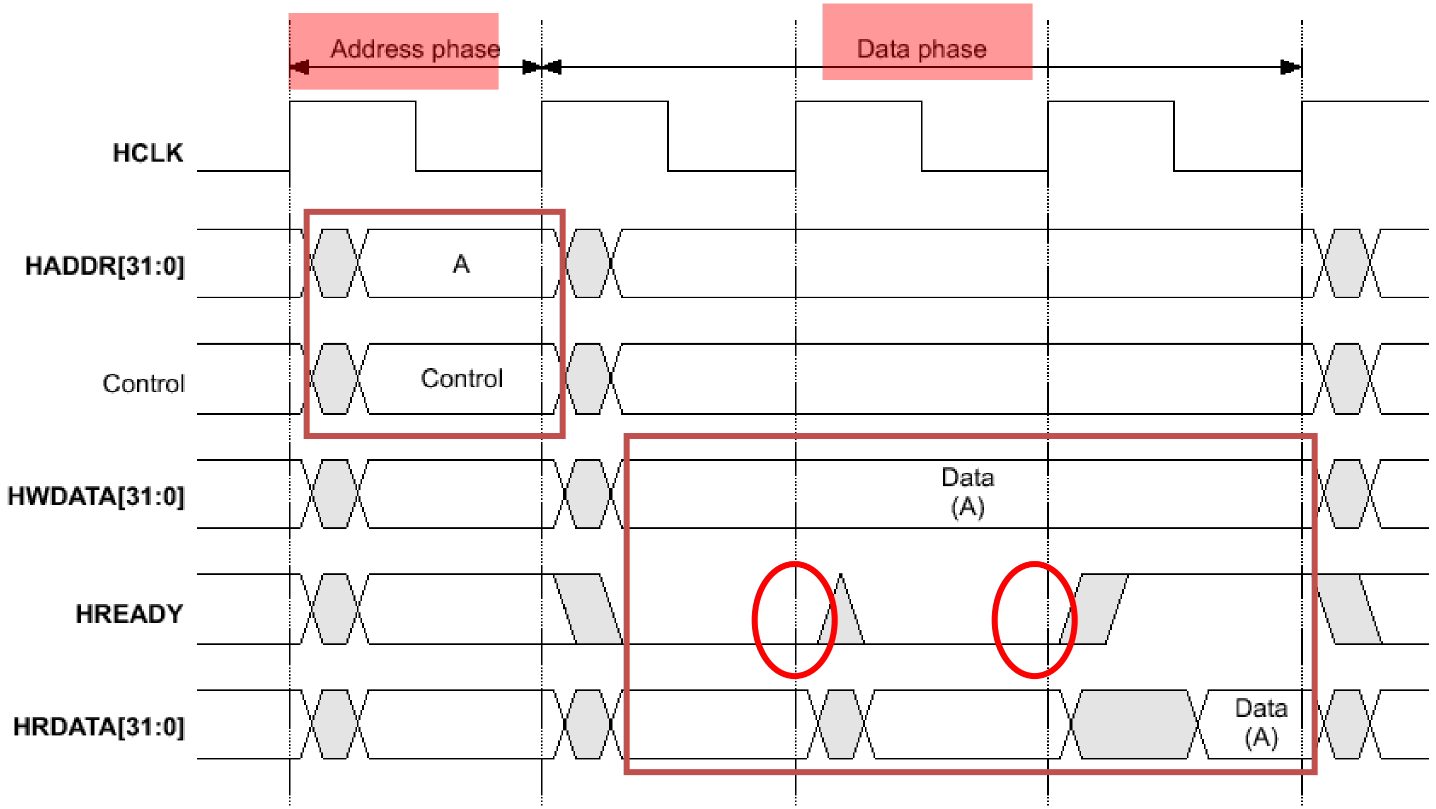


- **HCLK:** Reference for all AHB activity
- **HADDR:** 32 bit Address Bus
- **HWRITE:** Read/Write cycle
- **HWDATA:** Write Data Bus
- **HREADY:** Transfer complete response
- **HRDATA:** Read Data Bus

Two phases to complete an access as the pipeline
Address phase: lasts only a single cycle
Data phase: can be stretched for several cycles by slave

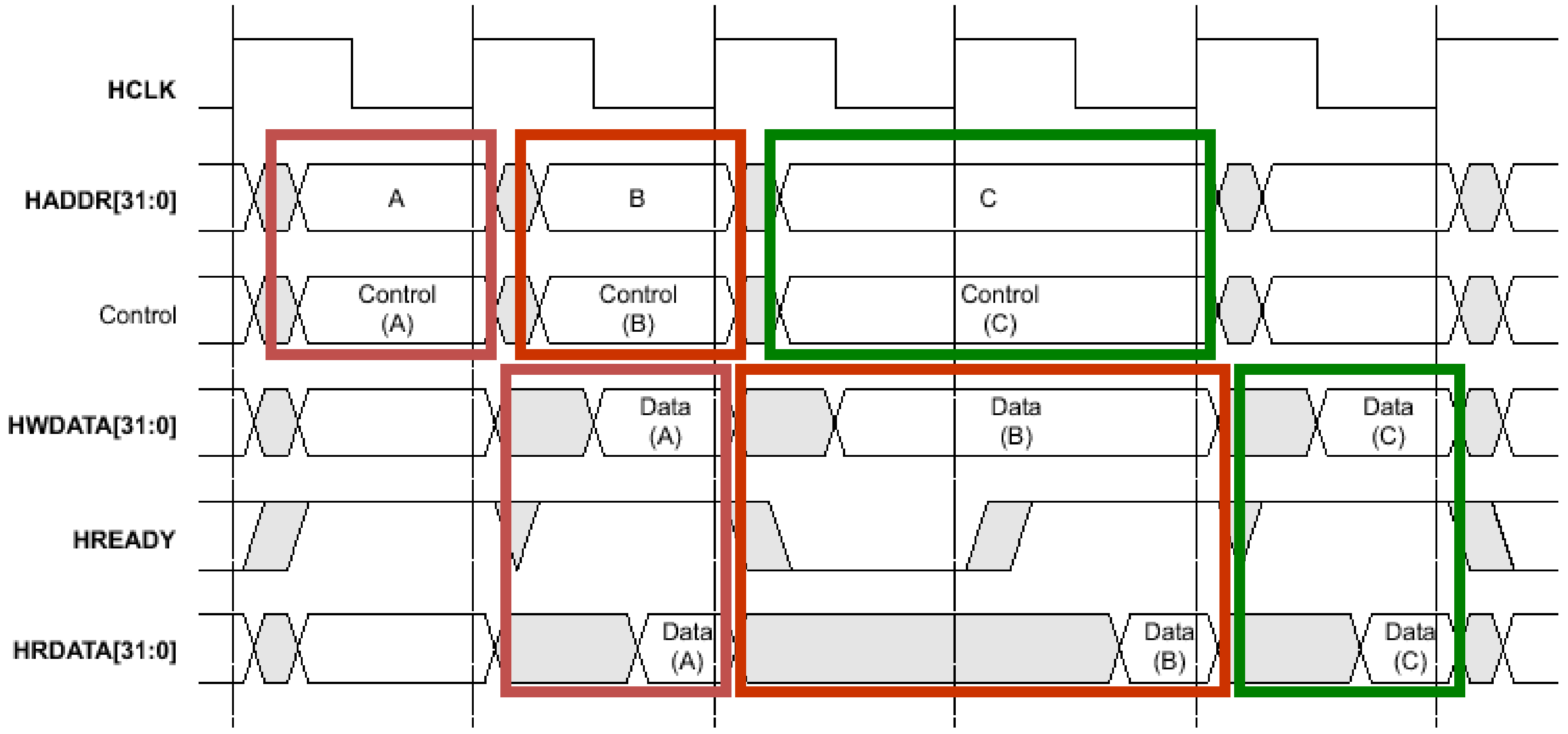


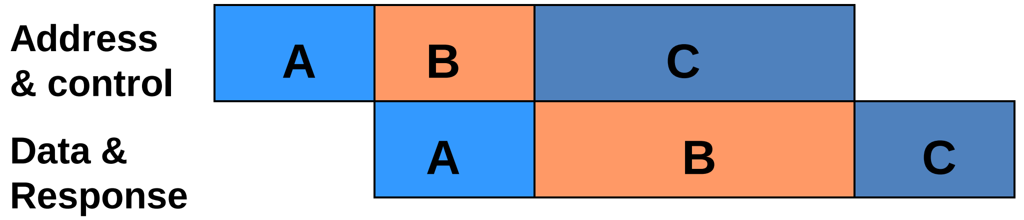
AHB Transfer with Wait States



* Access is stretched by slave which keep HREADY deassert
* It is recommended, but not mandatory, that slave do not insert more than 16 wait states

Multiple Transfers

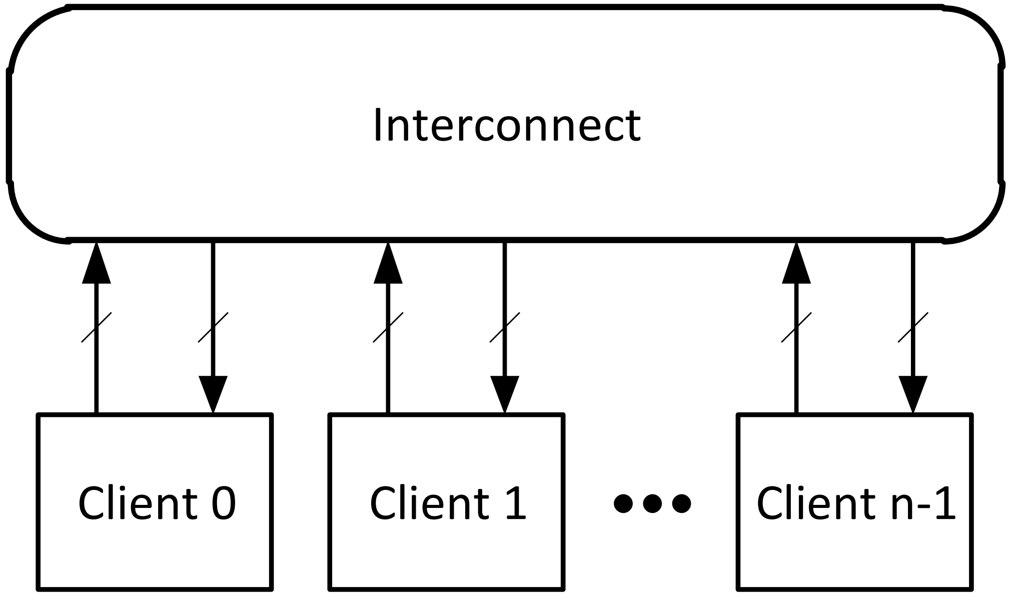




* Address phase of **C** is stretched because of stretched data phase of **B**

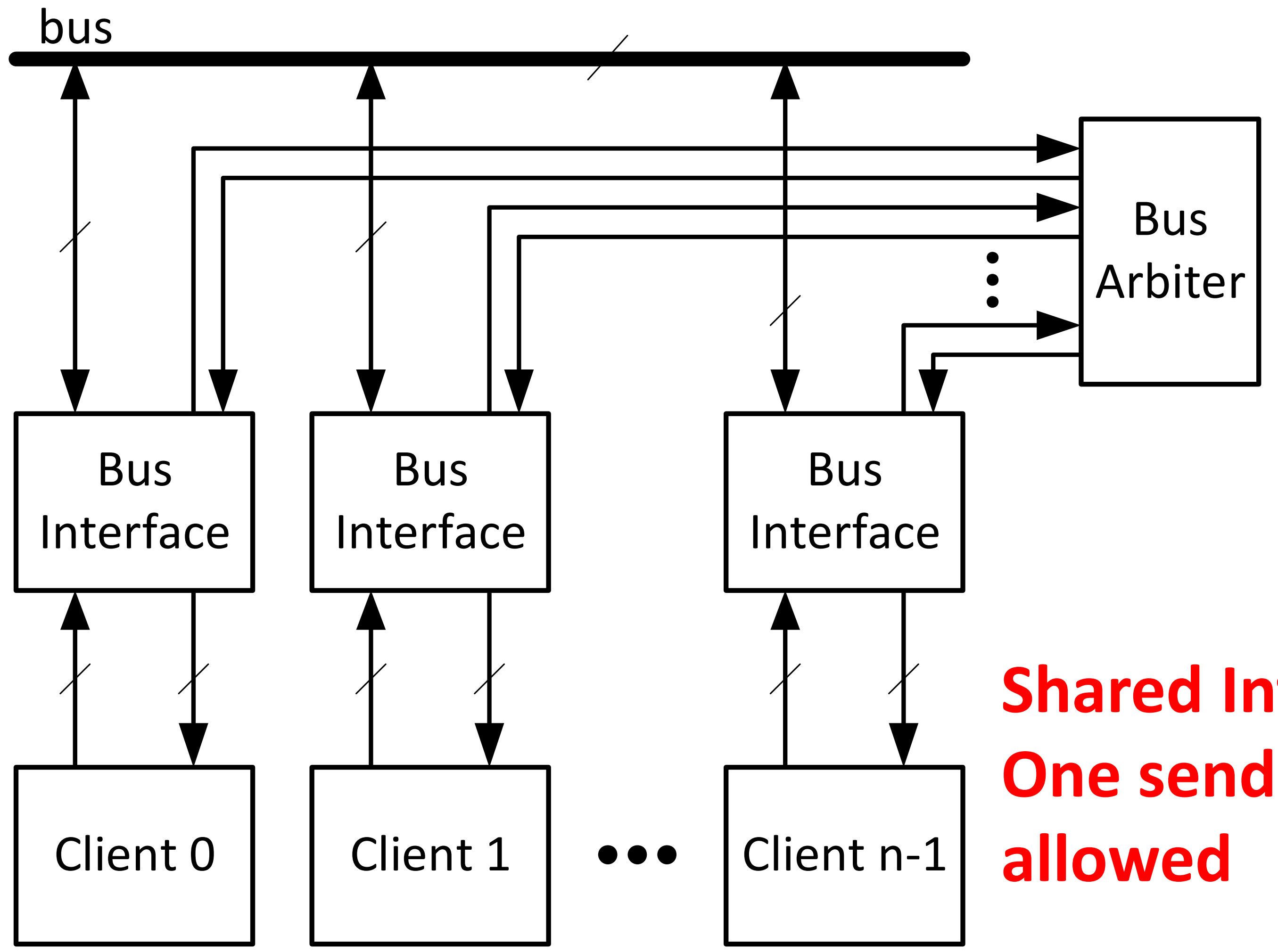
INTERCONNECTION

Interconnect 不只是一對一連接時，還能怎麼做?



- Many clients need to communicate
- Ad-hoc point-to-point wiring or shared interconnect
- Like a telephone exchange

1. Bus:

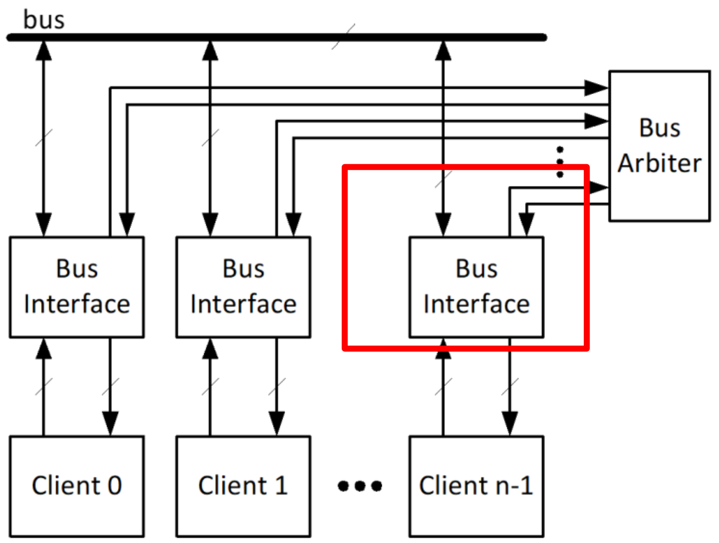


Shared Interconnect
One sender/receiver pair allowed

Verilog for a simple bus interface

```
// Combinational Bus Interface
// t (transmit) and r (receive) in signal names are from the perspective of the bus
module BusInt(cr_valid, cr_ready, cr_addr, cr_data, // bus rx - to the bus
              ct_valid, ct_data,                    // bus tx - from the bus
              br_addr, br_data, br_valid,           // to the bus
              bt_addr, bt_data, bt_valid,           // from the bus
              arb_req, arb_grant,                   // the arbiter
              my_addr) ;                            // address of this interface

  parameter aw = 2 ; // address width
  parameter dw = 4 ; // data width
  input cr_valid, arb_grant, bt_valid ;
  output cr_ready, ct_valid, arb_req, br_valid ;
  input [aw-1:0] cr_addr, bt_addr, my_addr ;
  output [aw-1:0] br_addr ;
  input [dw-1:0] cr_data , bt_data ;
  output [dw-1:0] br_data, ct_data ;
```

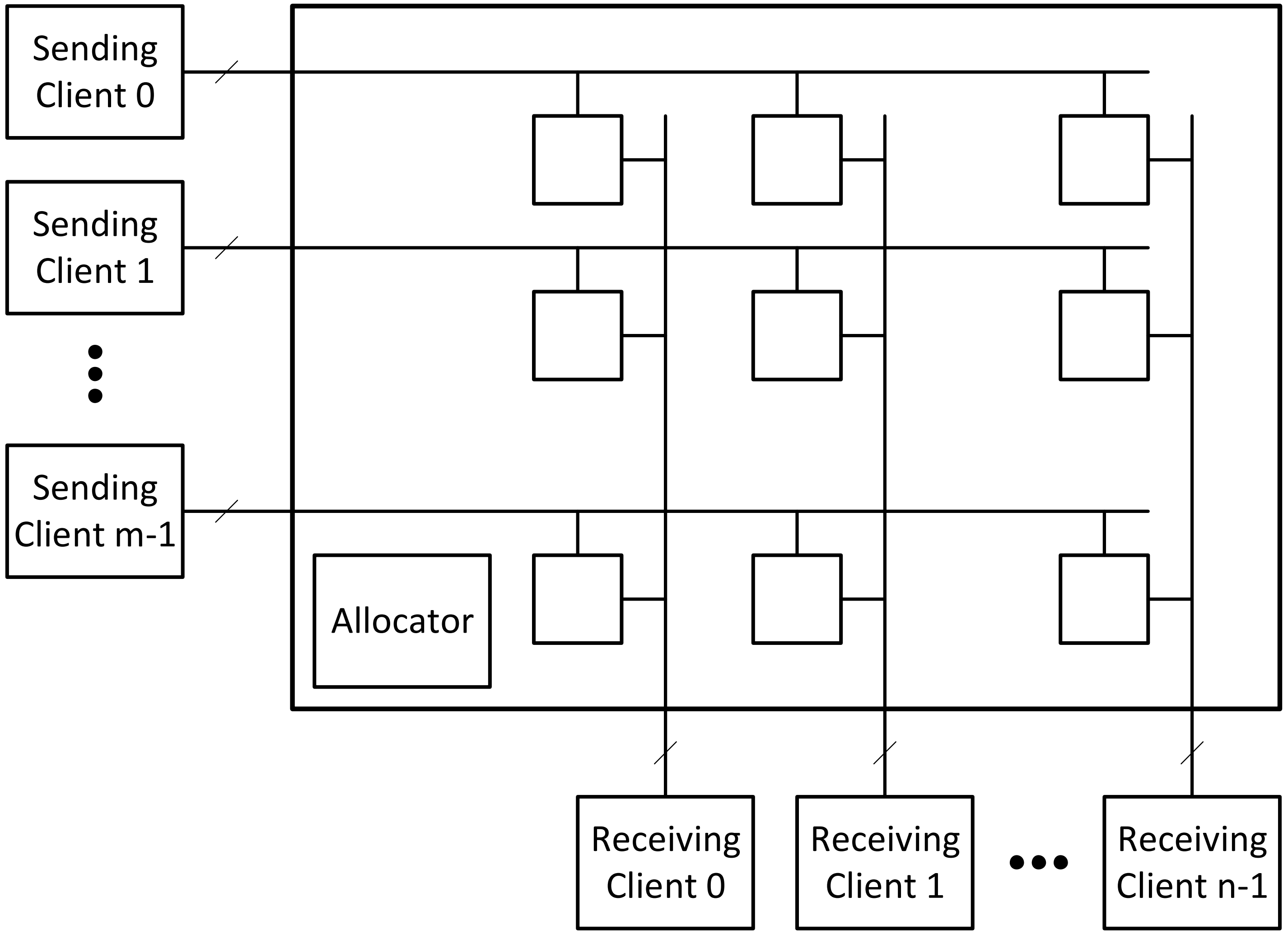


```
// arbitration
 wire arb_req = cr_valid ; //client data valid
 wire cr_ready = arb_grant ;//client ready

 // bus drive
 // assumes bus ORs these signals with those from other interfaces
 wire br_valid = arb_grant ; //client get grant to send data
 wire [aw-1:0] br_addr = arb_grant ? cr_addr : 0 ; //address
 wire [dw-1:0] br_data = arb_grant ? cr_data : 0 ; //data

 // bus receive
 wire ct_valid = bt_valid & (bt_addr == my_addr) ; //if bus address==myaddr,
 wire [dw-1:0] ct_data = bt_data ;
endmodule
```

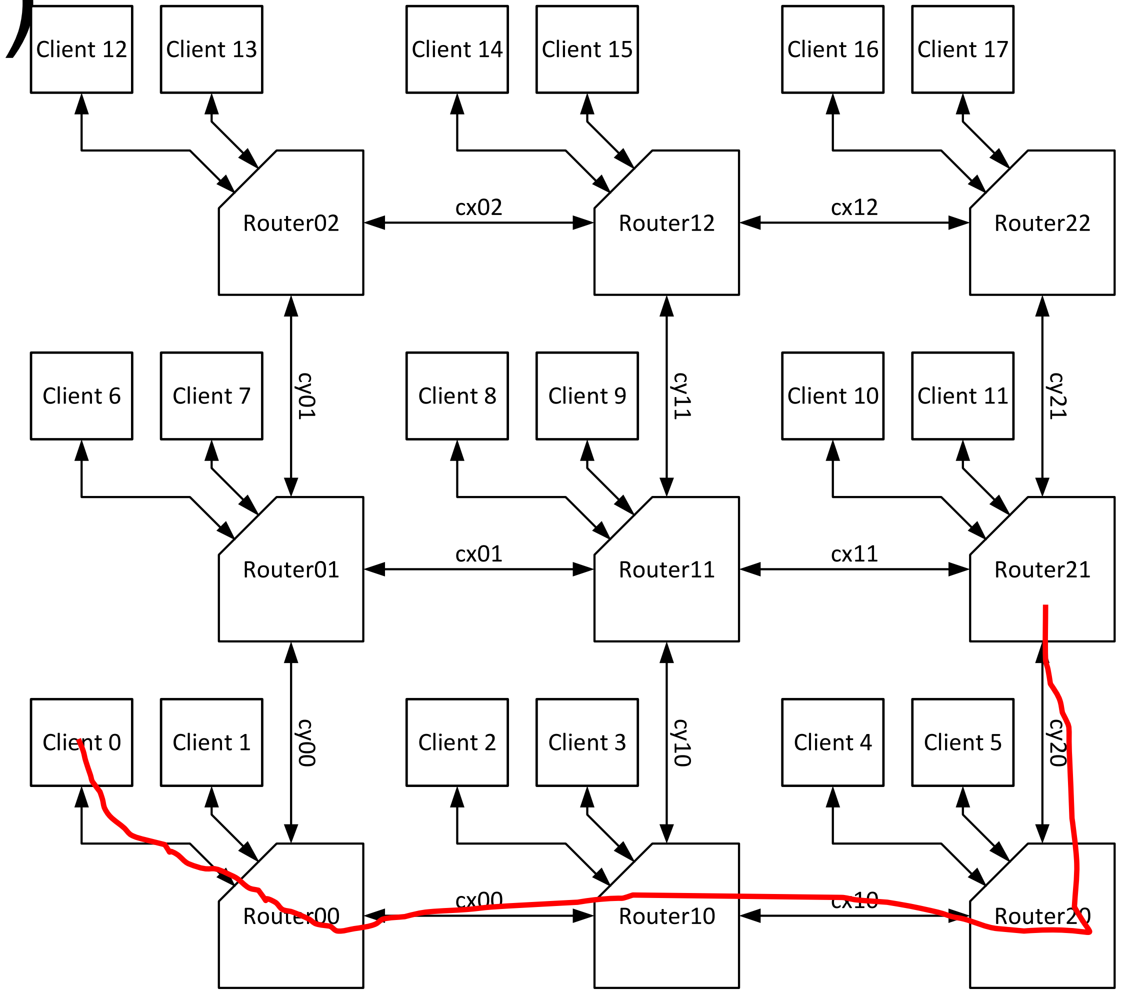
2. Crossbar Switch:



(c) 2005-2012 W. J. Dally

全連接型態
可同時多個傳送
只要目的不同
Multiple sender/receiver pairs

3. Interconnection Networks (Network on Chip: NoC)



網路型
可以容忍更長時間延遲

(c) 2005-2012 W. J. Dally

Interconnection Networks

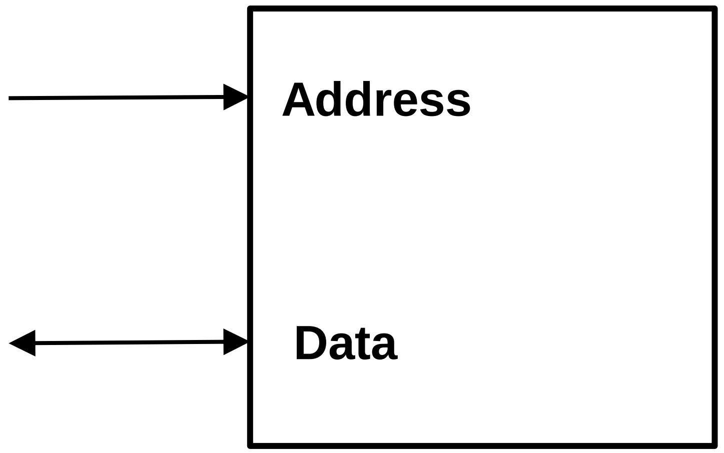
Cycle	0	1	2	3	4	5	6	7	8	9	10	11
Client 0 output	▭											
Router00		▭										
cx00e			▭									
Router10				▭	▭	▭						
cx10e							▭					
Router20								▭	▭			
cy20n										▭		
Router21											▭	
Client 11 input												▭

What factors determine which interconnect solution you pick?

- Bandwidth scalability
  - NoC > crossbar > bus
- Cost
  - Bus < crossbar < NoC

MEMORY

Memory



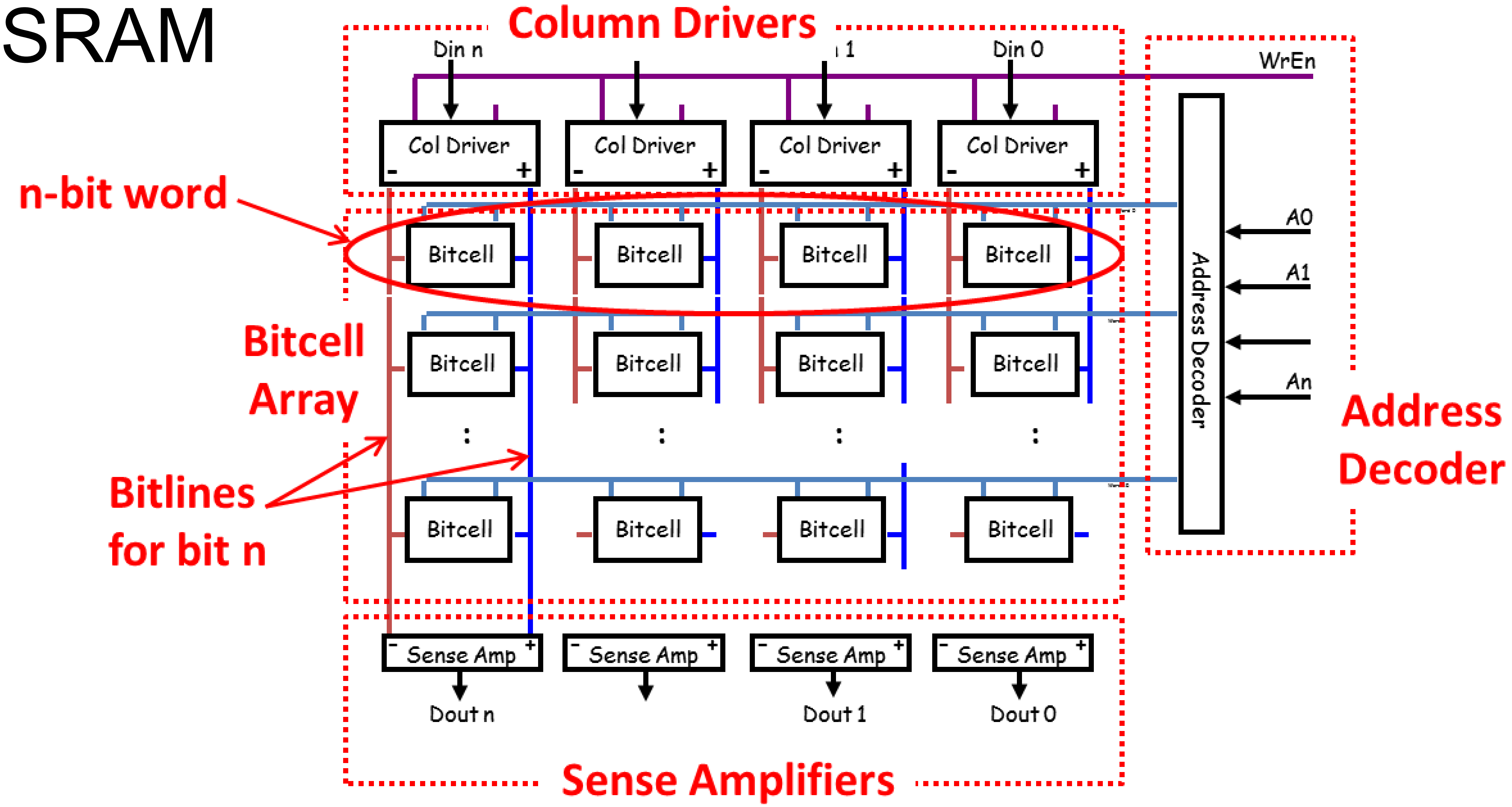
Capacity

Bandwidth

Latency

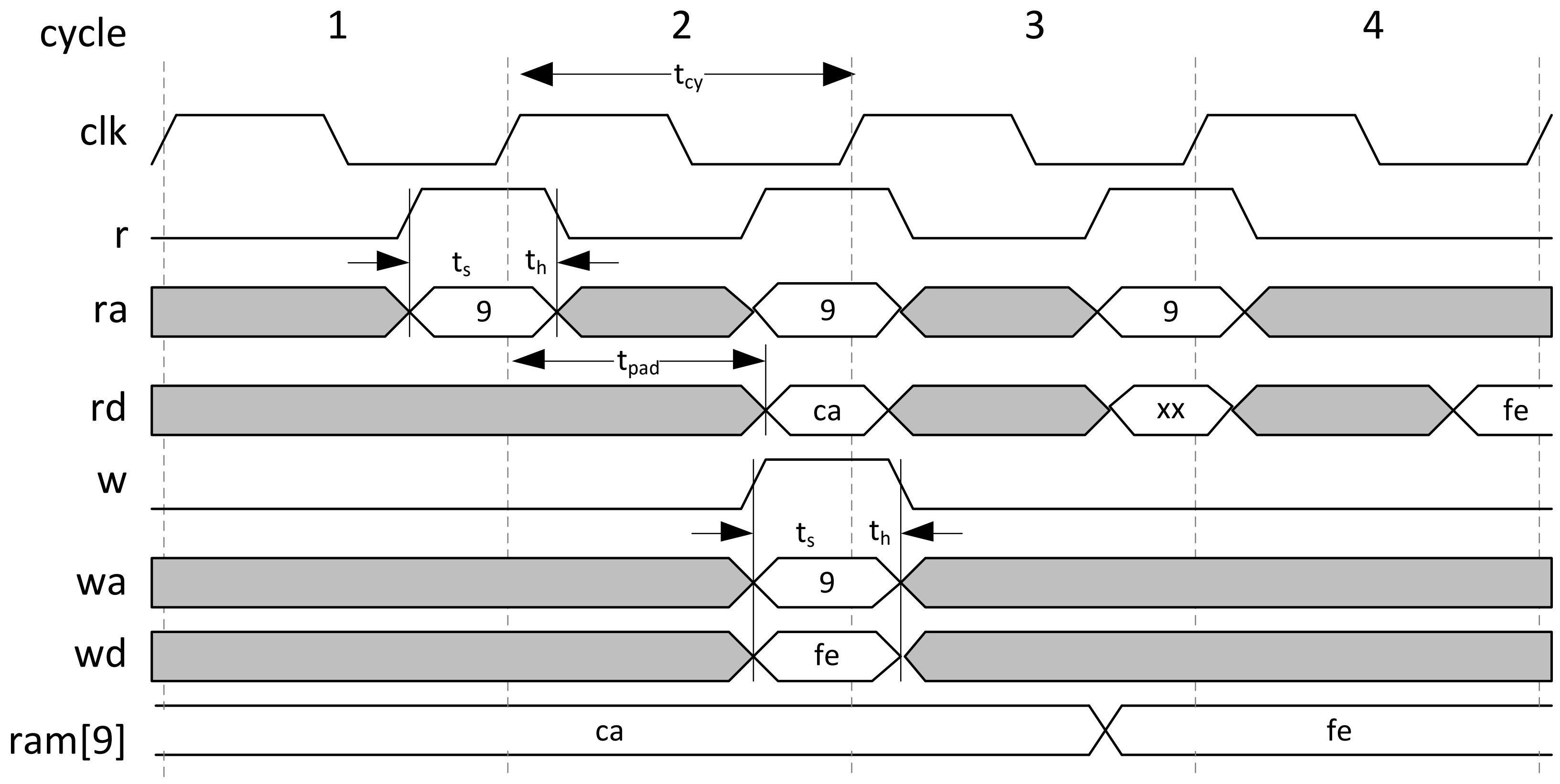
Granularity

SRAM

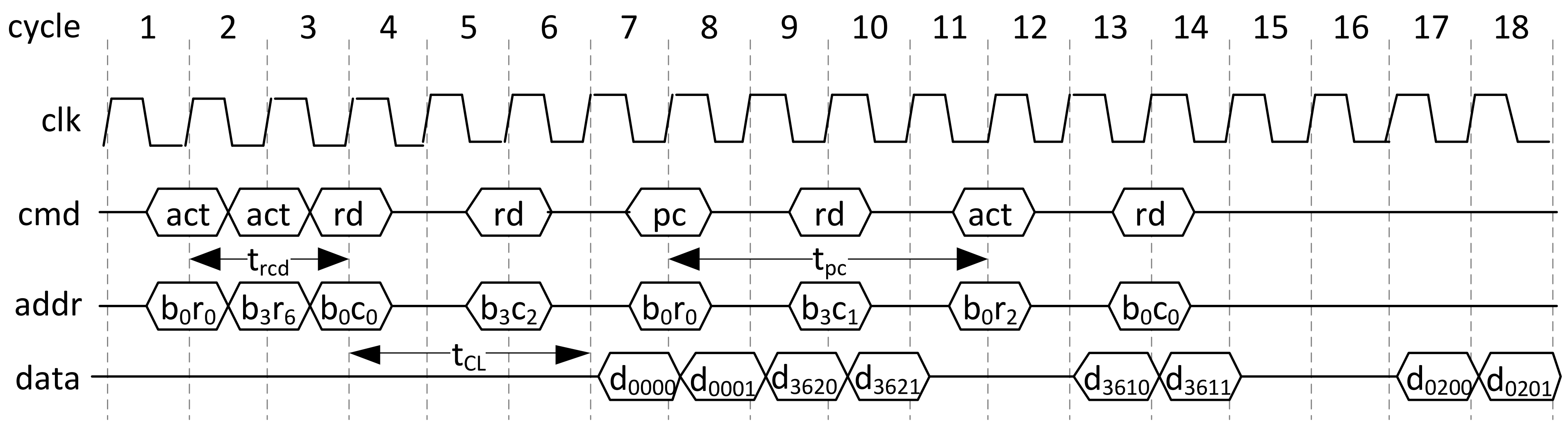


https://semiengineering.com/embedded-memory-impact-power-grids/

SRAM Primitive



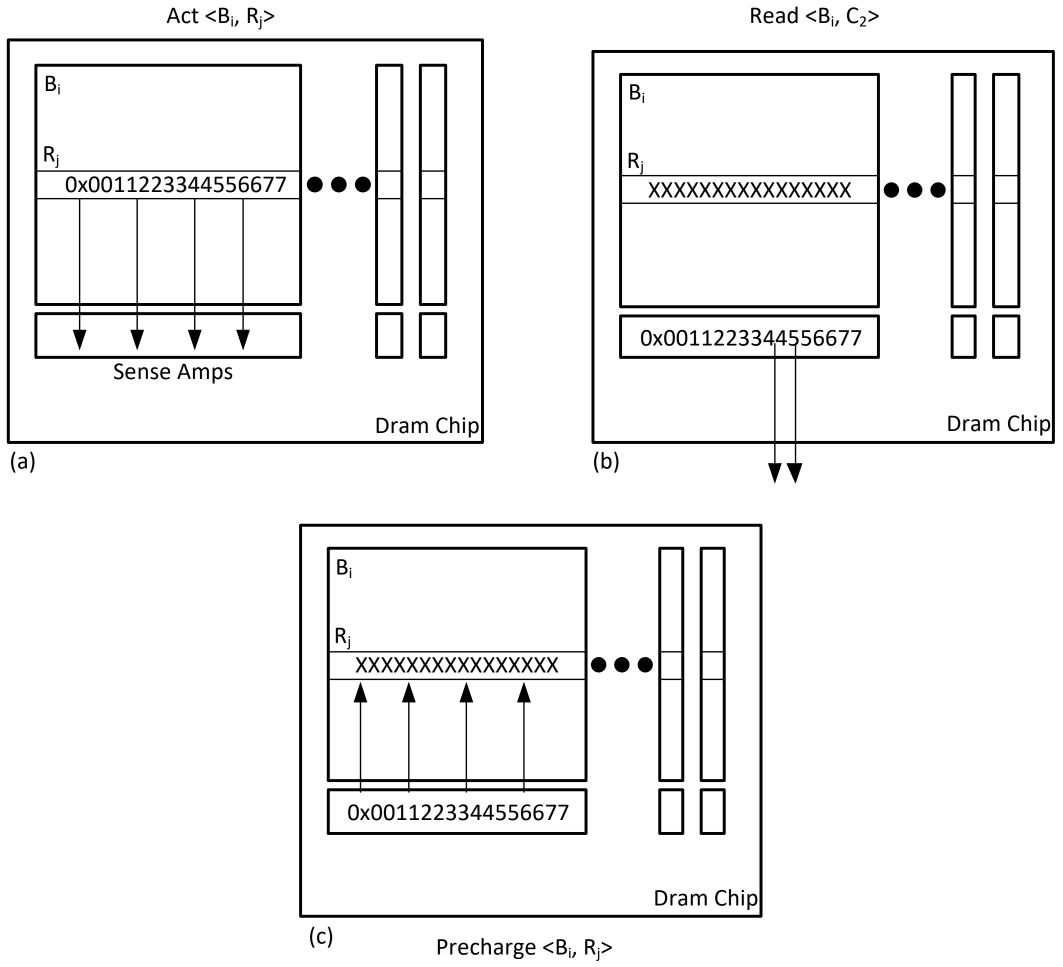
DRAM Primitive



Address is split into Bank, row, column
三步驟存取: bank/row activation, column access, precharge
同一bank，changed to 不同row要先precharge

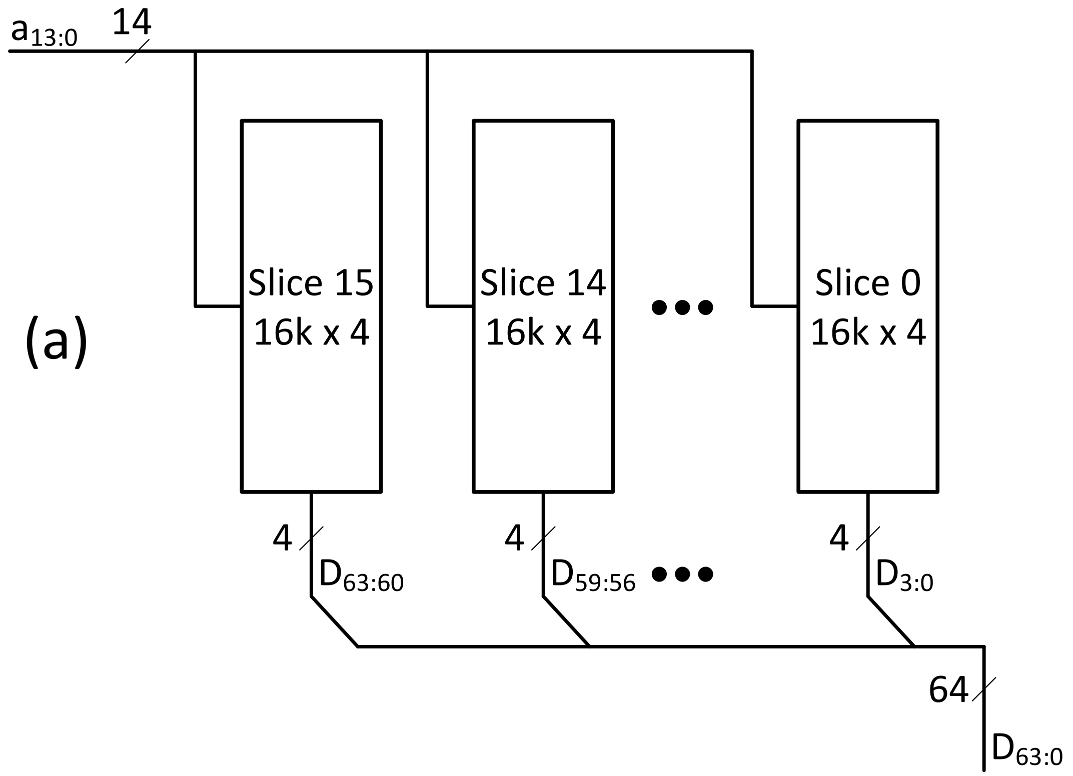
(c) 2005-2012 W. J. Dally

DRAM Operation



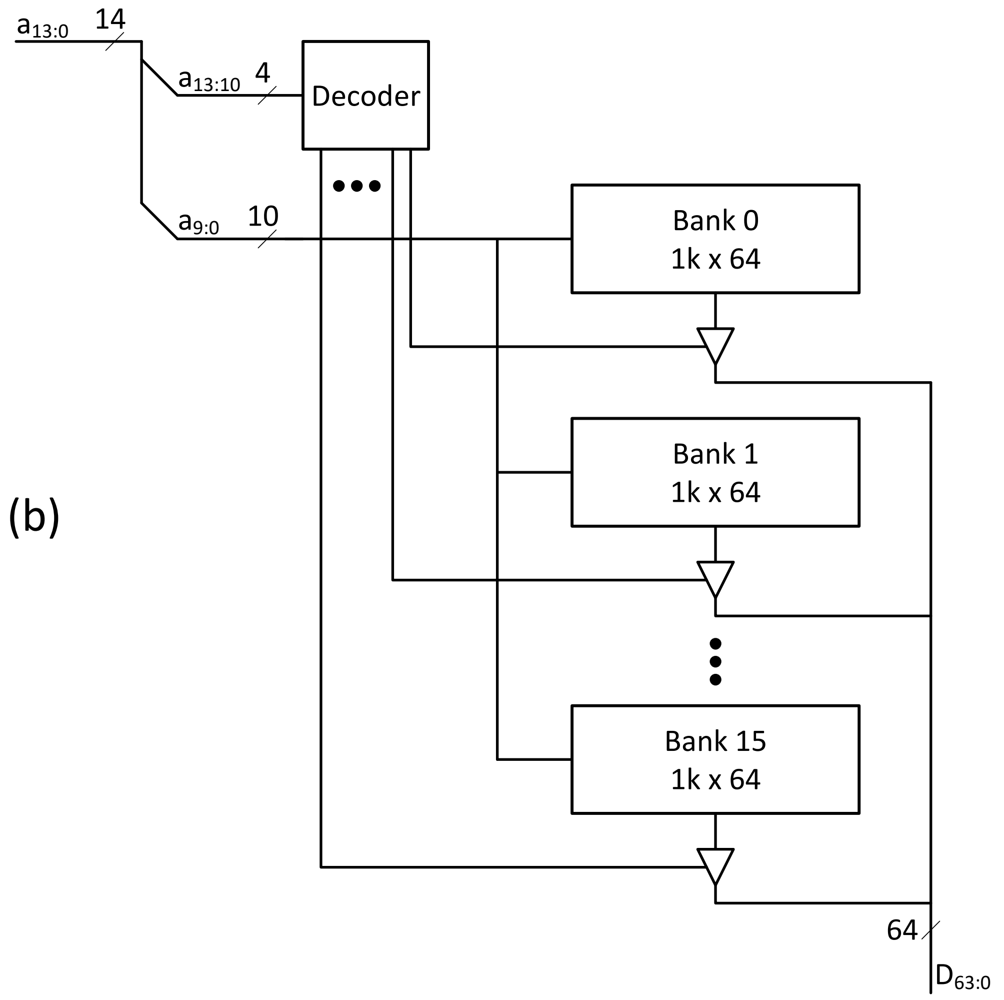
如果一塊MEMORY大小不夠或 BANDWIDTH輸出入頻寬不夠，那怎麼辦?
一塊不夠，那就兩塊，或更多塊

增加容量: Bit-Slicing



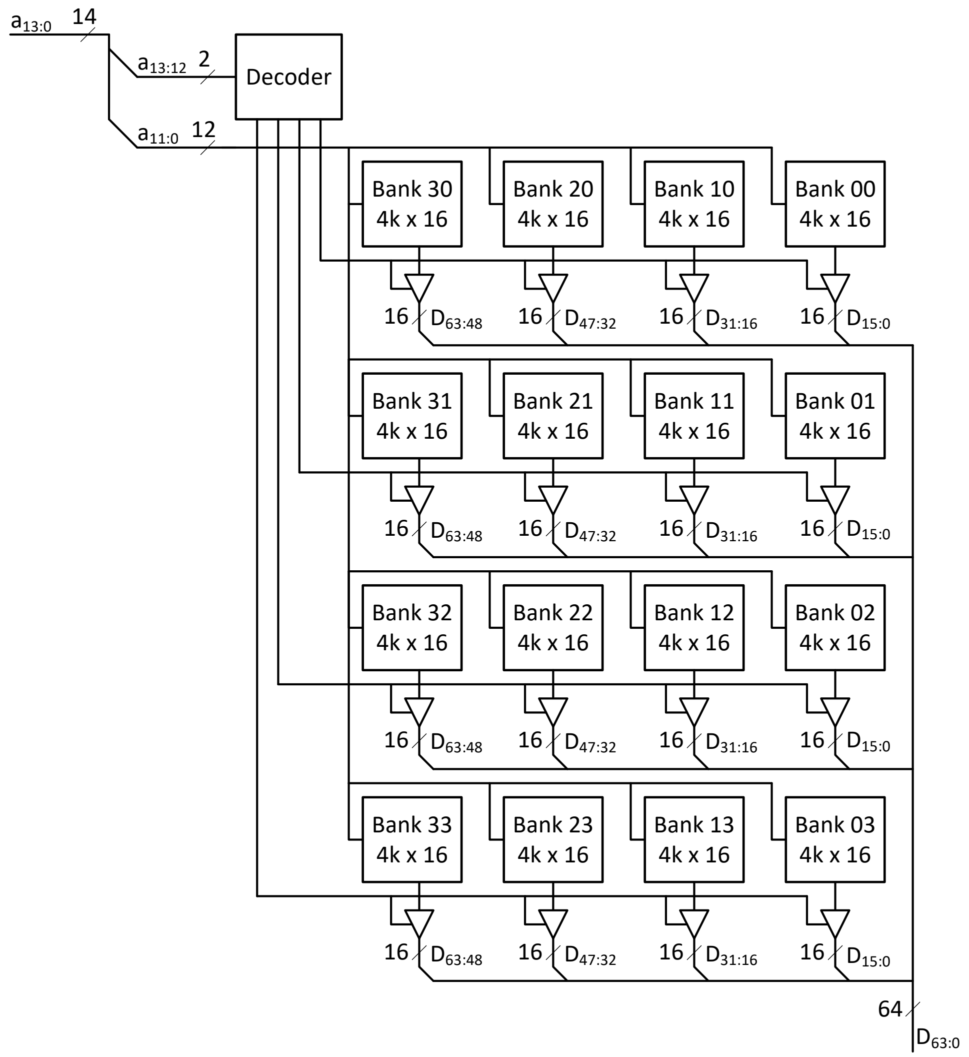
輸出入bitwidth可以增加
4bit x 16 => 64bit

增加容量: Banking

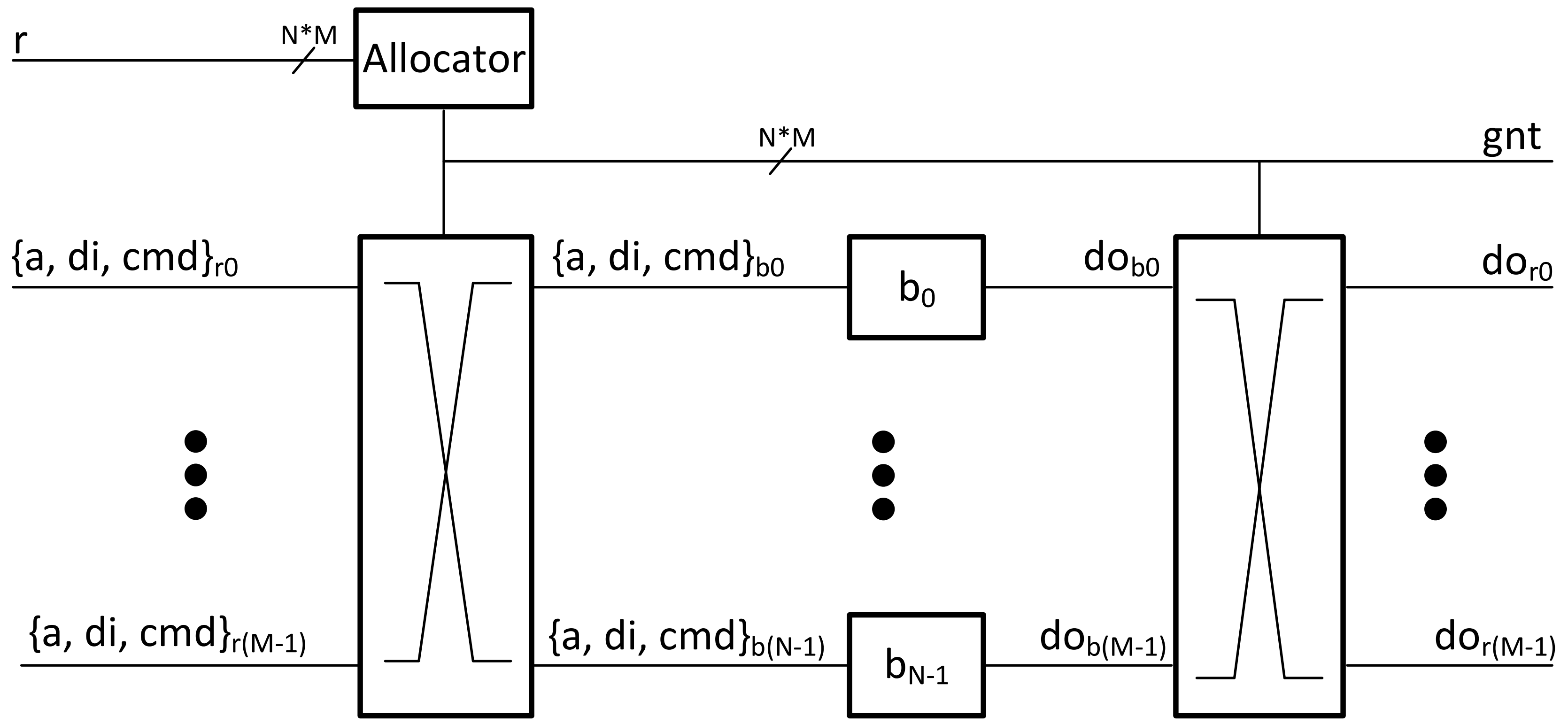


一次只讀寫一個，其他可以關掉省電

Bit slicing & banking



多個同時讀寫Interleaving



M request to N memory bank

Multiple bank 另種用法: 讀寫多個，但位址不一樣，有彈性

Avoid head-of-line blocking

Time	Q0	Q1	Q2	Q3	G0	G1	G2	G3
1	0,1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q0	–	–	–
2	1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q1	Q0	–	–
3	2,3	1,2,3	0,1,2,3	0,1,2,3	Q2	Q1	Q0	–
4	3	2,3	1,2,3	0,1,2,3	Q3	Q2	Q1	Q0
5	–	3	2,3	1,2,3	–	Q3	Q2	Q1
6	–	–	3	2,3	–	–	Q3	Q2
7	–	–	–	3	–	–	–	Q3

Table 24.2: With head of line blocking, the memory system does not have full throughput. Requests that can be granted are stuck behind requests waiting for a over-subscribed resource.

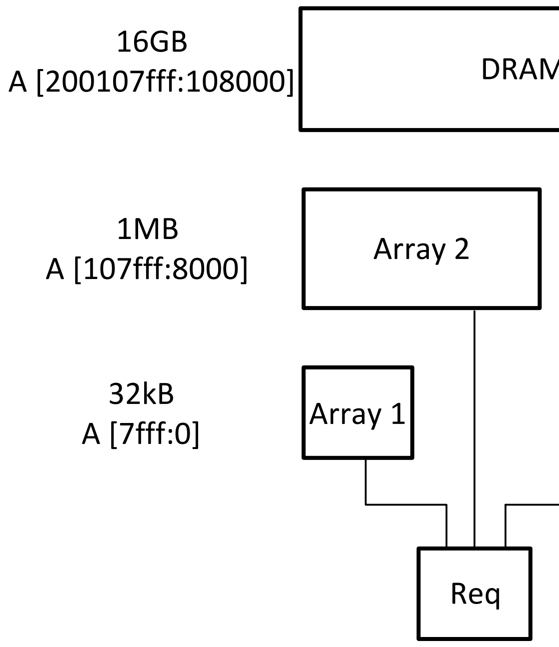
若只能照順序讀寫，會被卡住
使用率低

Time	Q0	Q1	Q2	Q3	G0	G1	G2	G3
1	0,1,2,3	0,1,2,3	0,1,2,3	0,1,2,3	Q0	Q1	Q2	Q3
2	1,2,3	0,2,3	0,1,3	0,1,2	Q3	Q0	Q1	Q2
3	2,3	0,3	0,1	1,2	Q2	Q3	Q0	Q1
4	3	0	1	2	Q1	Q2	Q3	Q0

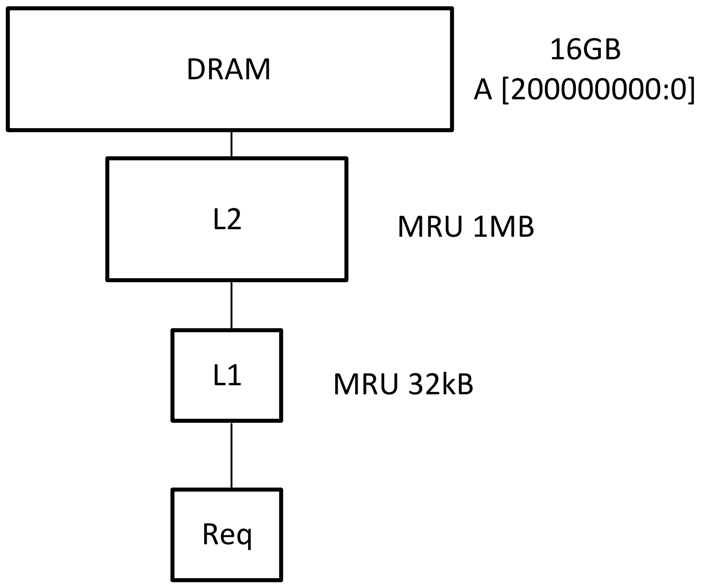
Table 24.3: When the arbiter is able to see requests beyond the head of the full, the memory system has no head of line blocking. It achieves full throughput.

不照順序讀寫，誰有空就去做
不會被卡住，使用率100%

Hierarchy



(a)



(b)

容量小的速度快，放在靠近logic附近，容量大的速度慢，放在比較遠的地方
可以搭配cache

Summary

- Interface timing
  - Convention about when data is transferred
    - When valid, when accepted
    - Open loop – always valid or periodic
    - Ready-valid flow-control, both ways, push, or pull
  - Serialization

- Interconnect
  - Allow any pair of clients to communicate
  - Bus, Crossbar, Network

- Memory
  - Capacity, bandwidth, latency, and granularity
  - SRAM and DRAM primitives
  - Bit-slice bank, or interleave to combine primitives
  - Hierarchy