Digital Design of Communication Systems

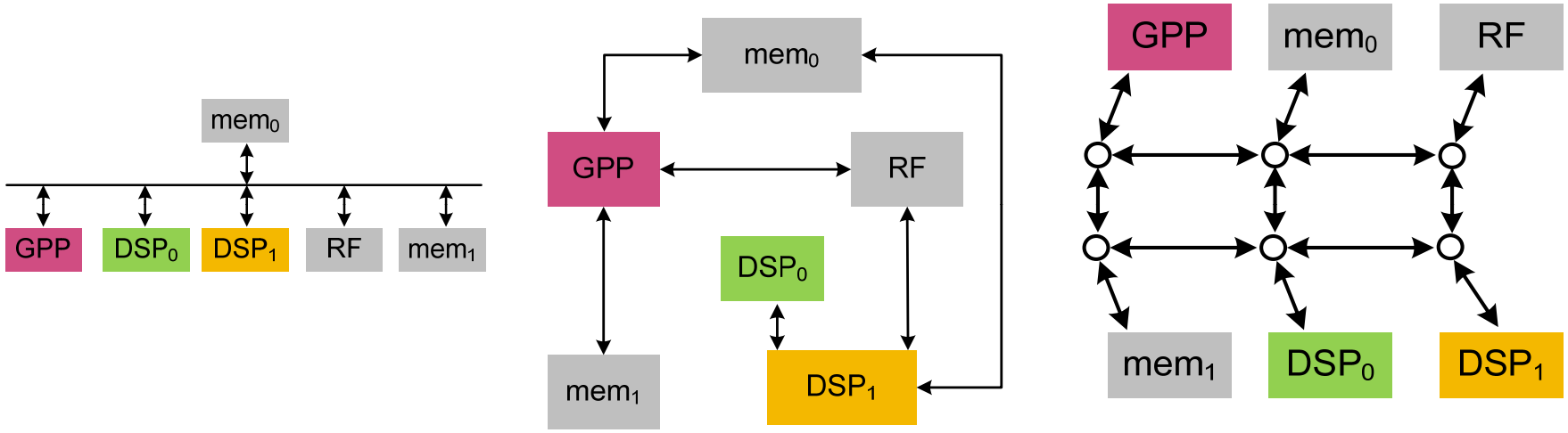
Lecture 13

Dr. Shoab A. Khan

Top-level Design Options

- A high data-rate communication system usually consists of hybrid components comprising ASICs, FPGAs, DSPs and GPPs
- A signal processing application requires sequential processing of a stream of data where the data input to one block is processed and sent to the next block for further processing
- The algorithm that runs in each block has different processing requirements
- There are various design paradigms used to interconnect components

Top-level Design Options



(a) (b) (c)

Interconnection configurations (a) Shared bus-based design. (b) Peer-to-peer connections (c) NoC-based design

Bus-based Design

- Shared bus
- System-on-chip (SoC) with multiple processing elements (PEs)
- A shared bus connects all the PEs in an SoC to a single bus
- The long bus has a very high load capacitance *C*
- This results in high power dissipation as power is directly proportional to *C* as given by the formula:

$$P = 1/2CfV^2$$

- where *f* and *V* are clock frequency and supply voltage, respectively

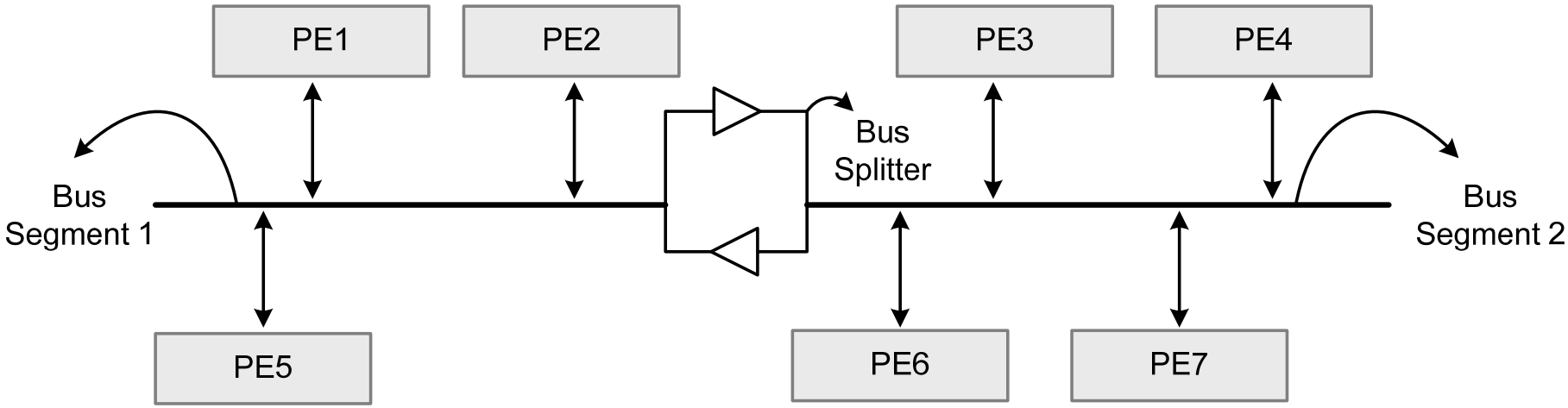
Segmented Bus-based Design

- Segmented shared bus design
  - Parallel transfers in multiple segments
  - As the length of each segment is small, so is its power dissipation

Bus-based Design

- Examples of on-chip shared-bus solutions:
  - AMBA (advanced microcontroller bus architecture) by ARM
  - SiliconBackplane µNetwork by Sonics
  - Core Connect by IBM
  - and Wishbone
- Example of a shared bus-based SoC : IBMCell processor
  - The bus provides four rings of connections for providing shared bandwidth to the attached computing resources
  - inherent limitation

Split shared bus-based design for a complex digital system



Point-to-Point Design

- Dedicated and exclusive interconnects
- Interconnection logic
- In design instances with deterministic inter-PE communication, and especially for communication systems where data is sequentially processed and only two PEs are connected with each other
- Attractive design option

Network-based Design

- All the components that need to communicate with other are connected to a network of interconnections
- Interconnections require short wires
  - Reduce power dissipation along with other performance improvements
- Intelligent network routes the data from any source to any destination in single or multiple hops of data transfers

Network-based Design

- This design paradigm is called network-on-chip (NoC)
  - easily accommodates multiple asynchronous clocked components
  - scalable solution as the number of cores on the SoC can be added using the same NoC design

Hybrid Connectivity

- A mix of three architectures can be used for interconnections
- A shared bus-based architecture: a good choice for applications that rely heavily on broadcast and multicast
- point-to-point architecture best suits streaming applications where the data passes through a sequence of Pes
- NoC-based interconnections best suit applications where any-to-any communication among PEs is required

Contd…

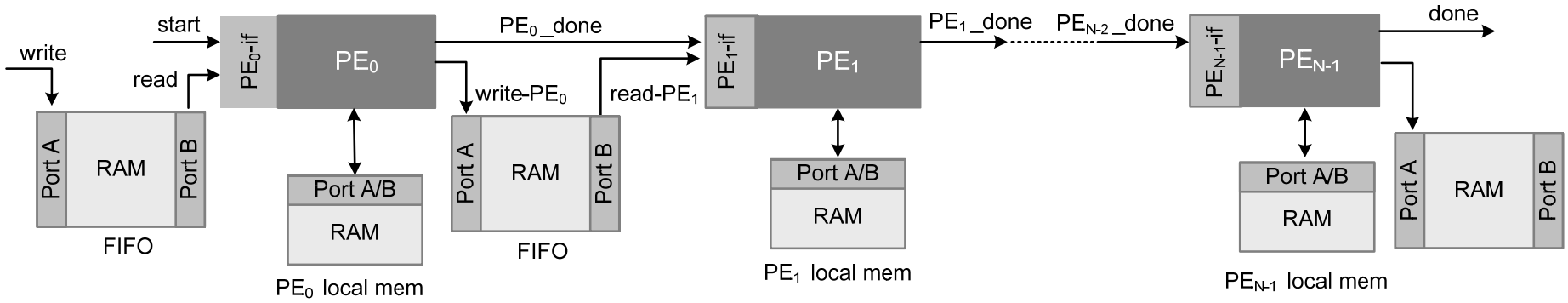
- Multiple programmable homogenous PEs that can execute a host of applications
- Multiple-processor SoC is also called an MPSoC
  - shared memories and external interfaces
  - Cisco Systems CRS-1 router; uses 188 extensible network processors per packet processing chip
- A hybrid of these three settings can be used in a single device

Point-to-Point KPN-based Top-level Design

- Good choice for mapping a digital communication system
- All PEs are either micro-program based architecture or time-shared architecture and run at asynchronous independent clocks, or
- A global clock where input to the system may be at the A/D converter sampling clock and the interface at the output of the system then works at the D/A converter clock
- A KPN-based design synchronizes these multi-rate blocks using FIFOs between every two blocks

KPN with Shared Bus and DMA Controller

- A direct memory access (DMA) is used to communicate with external memory and off-chip peripheral devices



KPN-based system for streaming applications

Contd…

- The peripheral devices and external memories are connected for data access
- All PEs have configuration and control registers mapped on GPP address space
- GPP first resets
- The micro-code is written using direct memory access to program memory of each PE
- When there are tables, then they are also DMA to data memories of specific PEs

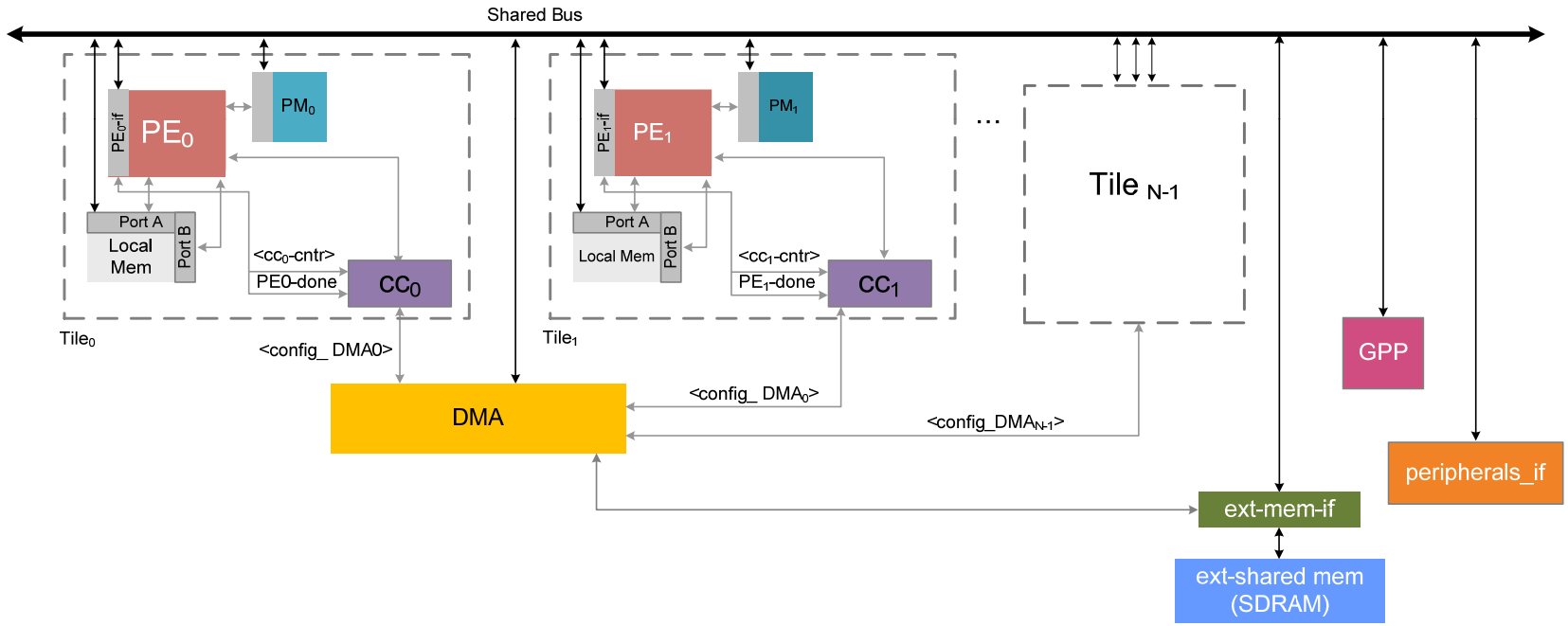
Contd…

- PE_i, after completing the task assigned to it, sets the PE_i _*done* flag
- When the next processing on the data is to be done by an external DSP, ASIC, or any other PE on the same chip, the done signal is sent to the respective CC_i
- This sets a DMA channel for data transfer

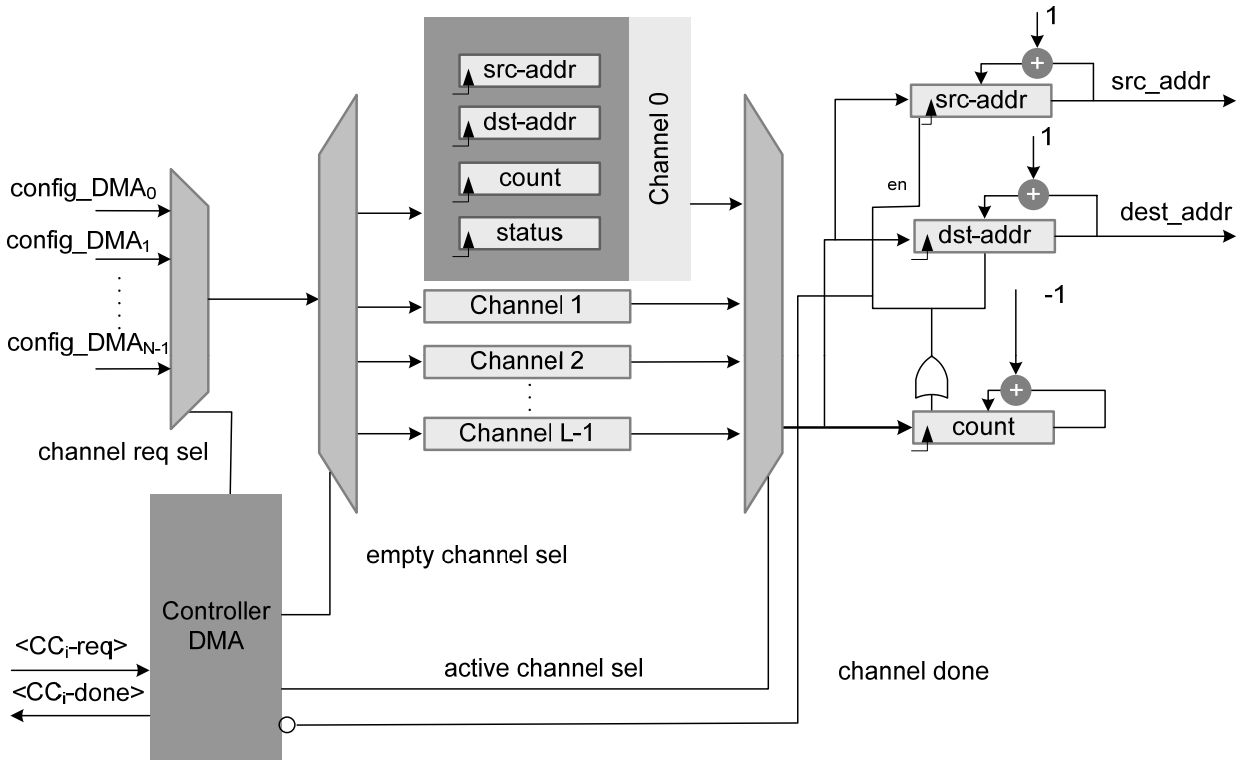
Contd…

- The DMA gets the access to the shared bus and makes the data transfer between on-chip and off-chip resources
- A PE can also set DMA to make data transfers from its local data memory to the DM of any other PE

Shared-bus and KPN-based top-level design with DMA for memory transfers



Representative design of a DMA



Multi –channel DMA design

Representative design of a DMA

- Multiple channels to serve all CCs
- Each CC_i is connected to DMA through the *config_DMA* field
- This has several configuration-related bits such as CC_i *_req*
- A request to DMA
- Processing of these requests by copying the source, destination and block size to an empty DMA channel

Contd…

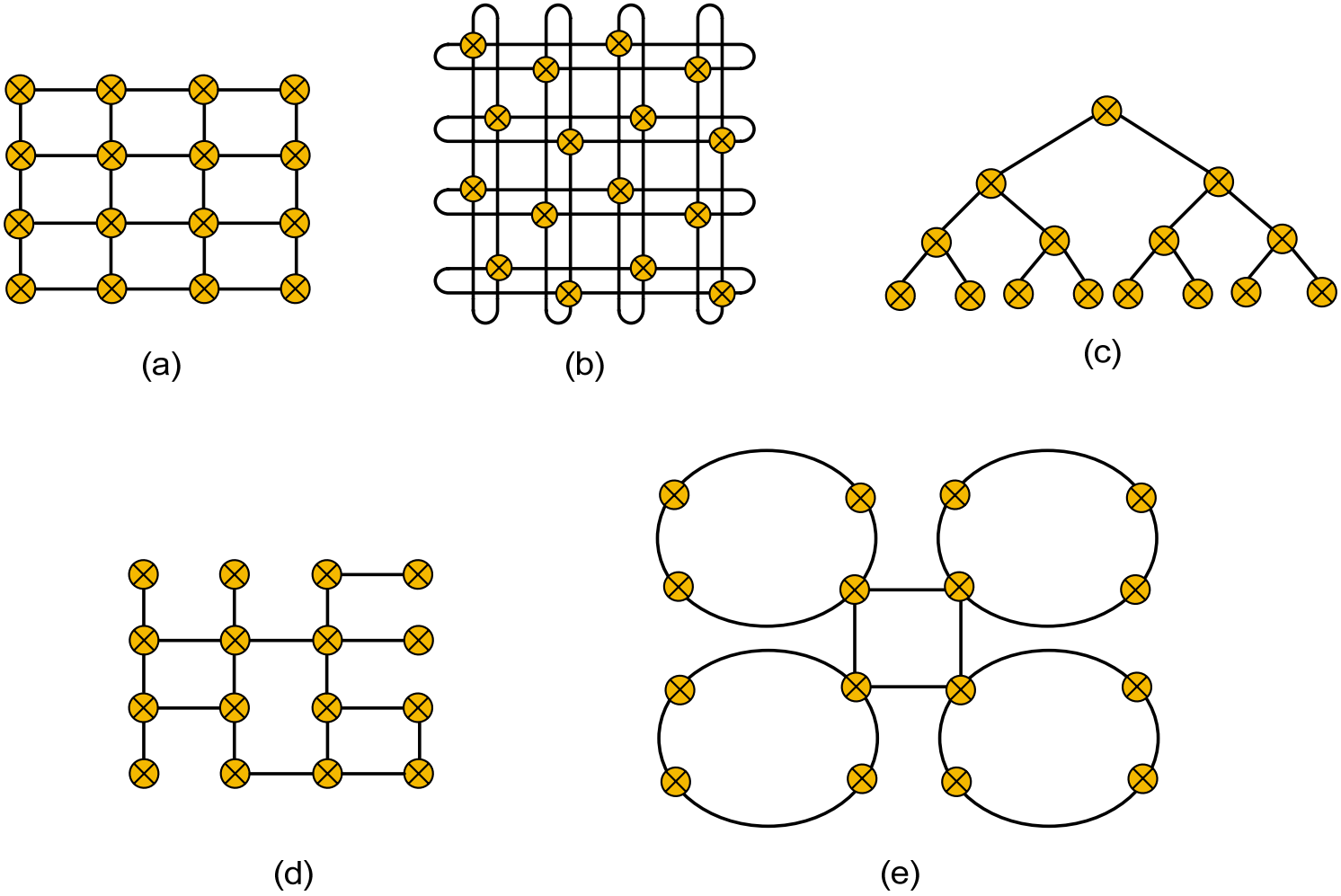
- The controller also, in a round-robin arrangement, copies a filled DMA channel to execute a DMA channel for actual data transfer
- The DMA gets access to the shared bus and then makes the requisite transfer by first bringing the data to its local buffer and then transferring it to the destination address
- After the DMA is done with processing the channel, the DMA assert CC_i*_req_done* flag and also set the DMA channel free and starts processing the next request if there is any

Network-on-Chip Top-level Design

- The NoC paradigm provides an effective and scalable solution to inter-processor communication in MPSoC designs
- The network topology and routing algorithms
- The cores of PEs may be homogenous
- In many application-specific designs, the PEs may not be homogenous, so they may or may not be connected in a regular grid
- To connect the PEs a number of interconnection topologies are possible

Topologies

- The physical structure and connections of nodes in the network



Network-on-chip configurations (a) grid; (b) torus (c) binary tree (d) irregular connectivity (e) mixed topology

Contd ...

- The final selection is based on
  - the traffic pattern,
  - the requirements of quality of service (QoS)
  - budgeted area
  - and power for interconnection logic
- Grid [8] and torus [6] arrangements of routers are shown in Figures (a) and (b)
- In these, a router is connected to its nearest neighbors. Each router is also connected with a PE core that implements a specified functionality.
- Best arrangement for homogenous PEs or a generic design that can be used for different applications
- run-time reconfigurable architectures

Switching Options

- Two switching options in NoC-based design:
  - circuit switching
  - packet switching
- Deals with transportation of data from a source node to a destination node in a network setting
- In circuit switching, a dedicated path from source to destination is first established
- Packet switching breaks the data into multiple packets. These are transported from source to destination

Store and Forward Switching

- In this setting, router A first requests its next-hop router B for transmission of a packet
- After receiving a grant, A sends the complete packet to B
- After receiving the complete packet, B similarly transmits the packet to the next hop, and so on

Virtual Cut-through Switching

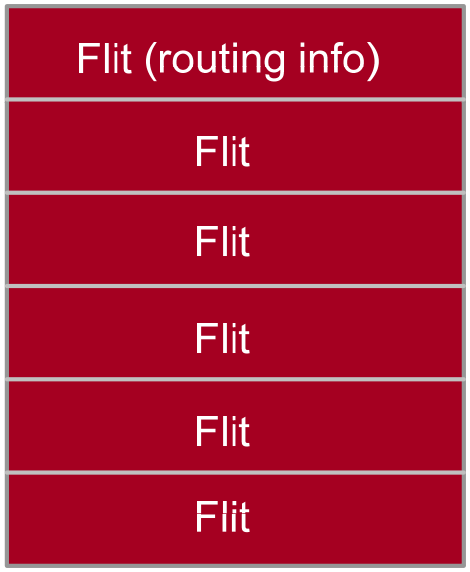
- In this setting, a node does not wait for reception of a complete packet
- Router A, while still receiving a packet, sends a request to the next-hop router B
- After receiving a grant from B, A starts transmitting the part of the packet that it has already received, while still receiving the remainder from its source
- To cover for contention, each node must have enough buffer to hold an entire packet of data

Wormhole Switching

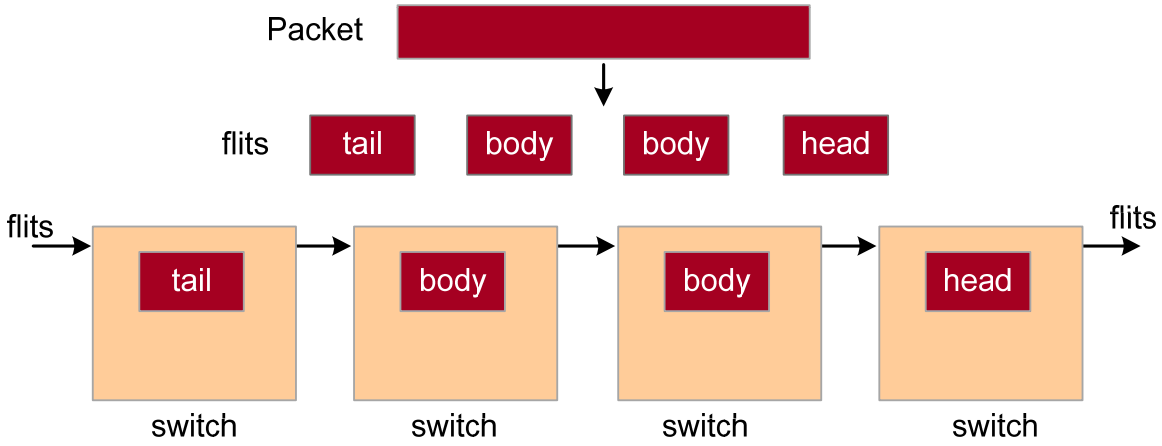
- In wormhole switching, the packet is divided into small data units called ‘flits’
- The first flit is the *head flit* and it contains the header information for the packet and for the flits to follow
- The data itself is divided into several *body flits*. The last flit is the *tail flit*
- Following figure shows division of a packet into flits and a snapshot of their distribution in a number of switches in a path from source to destination
- All routers operate on these smaller data units. In this switching the buffers are allocated for flits rather than for entire packets

Wormhole switching that divides a packet into multiple flits.

(a) Packet divided into flits.
(b) Wormhole switching of flits



(a)



(b)

Routing Protocols

- A router implements options out of a number of routing protocols
- All these protocols minimize some measure of delays and packet drops
- The delay may be in terms of the number of hops or overall latency
- In a router on a typical NoC, a switch takes the packet from one of the inputs to the desired output port
- Alternatively the system may implement routing where only the destination and source IDs are appended in the packet header.

Routing Protocols

- Alternatively the system may implement routing where only the destination and source IDs are appended in the packet header
- A router based on the destination ID routes the packets to the next hop
- For a more involved design where the PEs are processing voice, video and data simultaneously, a QoS based routing policy may also be implemented that looks into the QoS bits in each received packet and then prioritizes processing of packets at each router
- For the simplest designs, pre-scheduled routing may also be implemented
- An optimal schedule for inter-processor communication can then be established
- Based on the schedule, the NoC can be optimized for minimum area and power implementation

Routing Algorithms

- For a NOC, a routing algorithm determines a path from a set of possible paths that can virtually connect a source node to a destination node
- Three types of routing algorithm:
  - deterministic,
  - oblivious
  - and adaptive
- The *deterministic algorithm* implements a predefined logic without any consideration of present network conditions
  - This setting for a particular source and destination always selects the same path
  - May cause load imbalance and thus affects the performance of the NoC. The routing algorithm is, however, very simple to implement in hardware.

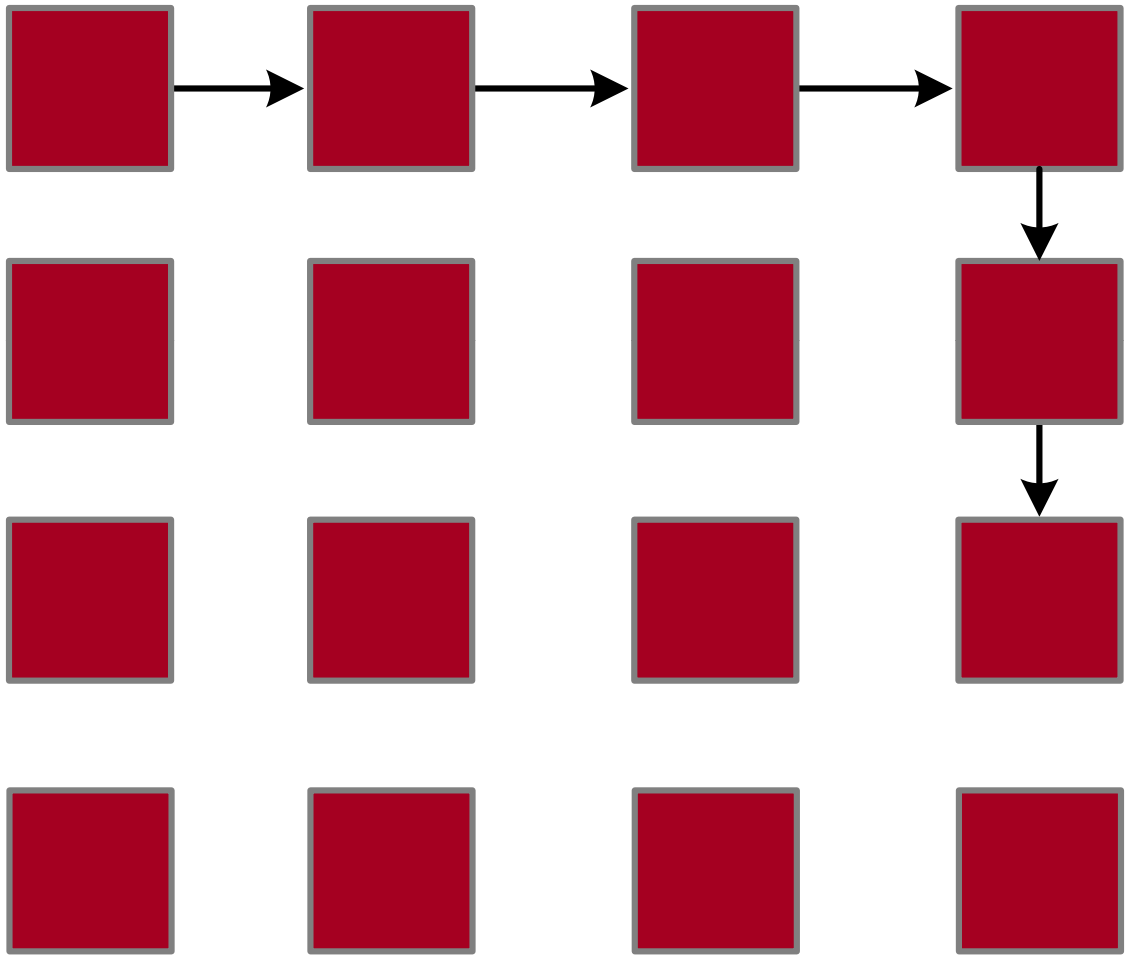
Routing Algorithms

- Deterministic algorithm
  - implements a predefined logic without any consideration of present network conditions
- Oblivious algorithm
  - A subset of all possible paths from source to destination is selected
  - The hardware selects a path from this subset in a way that it equally distributes the selection among all the paths
  - These algorithms do not take network conditions into account
- Adaptive algorithm
  - Includes the status of links and buffers and channel load condition
  - A routing decision at every hop can be made using a look-up table
  - An algorithm implanted as combinational logic may also makes the decision on the next hop

XY Routing

- A simple to design logic for routing a packet to the next hop
- The address of the destination router is in terms of (*x*,*y*) coordinates
- A packet first moves in the *x* direction and then in the *y* direction to reach its destination, as depicted in the following figure .
- For cases that result in no congestion, the transfer latency of this scheme is very low, but its packet throughput decreases sharply as the packet injection rate in the NoC increases.
- This routing is also deadlock free but results in an unbalanced load on different links.

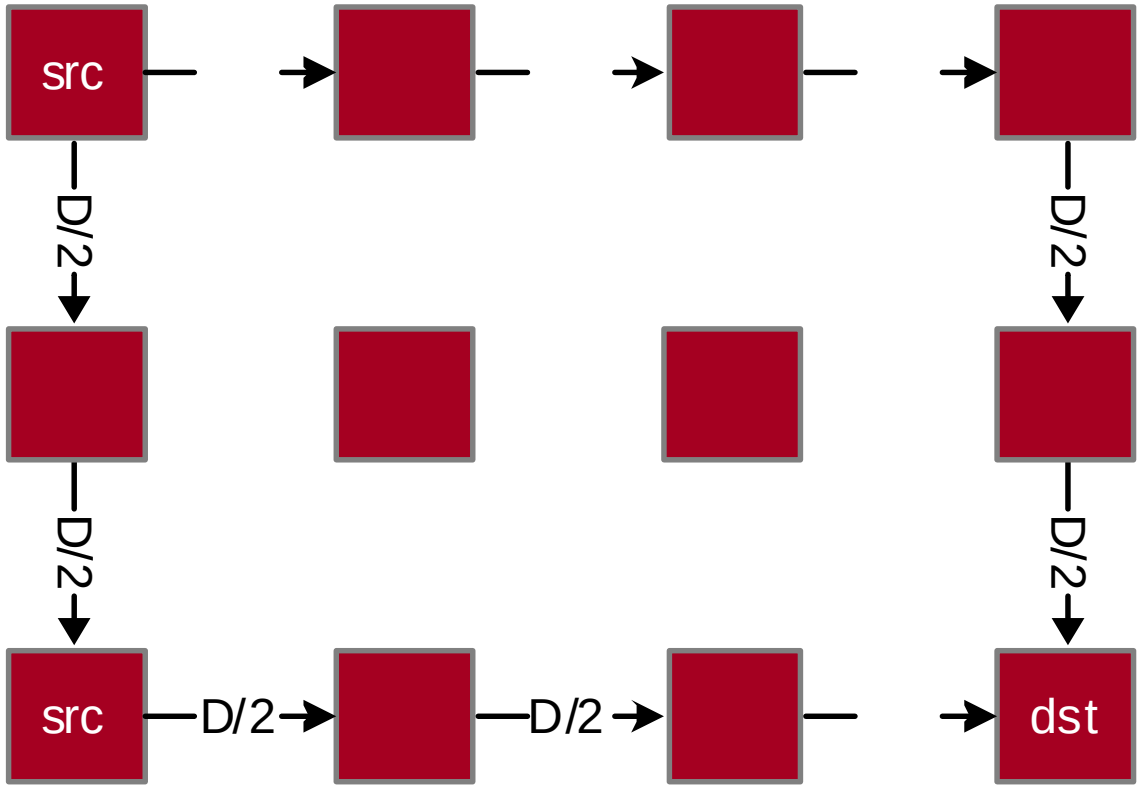
XY Routing



Toggled XY routing

- This is from the class of oblivious schemes
- It splits packets uniformly between XY and YX
- If a packet is routed using YX, it first moves in the *y* direction and then in the *x* direction
- It is also very simple to implement and results in balanced loads for symmetric traffic.
- Figure shows an NoC performing toggled XY routing by switching D/2 packets in XY and the others in YX while routing a uniform load of D packets

Toggled XY routing

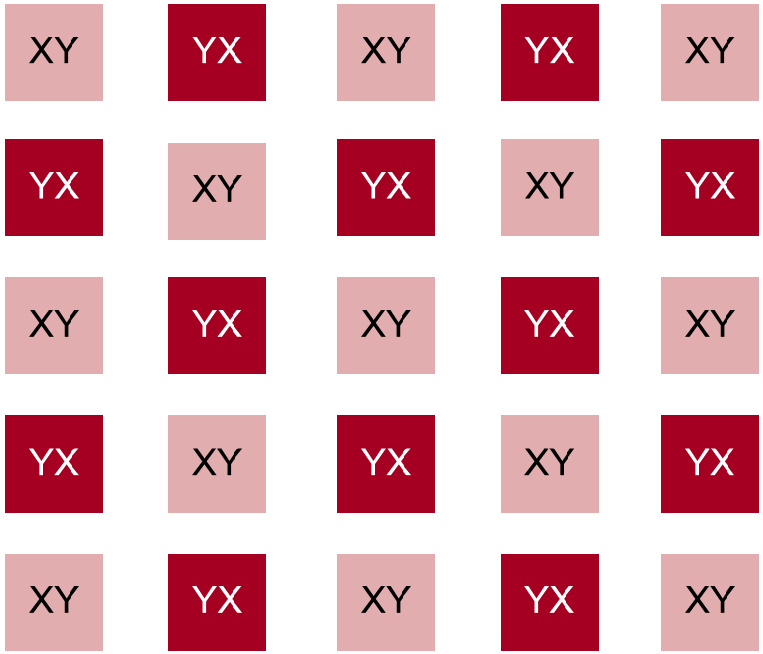


Weighted Toggled XY Routing

- The weighted version of toggled XY routing works on a priori knowledge of traffic patterns among nodes of the NoC. The ratios ofXYandYXare assigned to perform load balancing
- Each router has a field that is set to assign the ratios. To handle out-of-sequence packet delivery, the packets for one set of source and destination are constrained to use the same path

Source-Toggled XY Routing

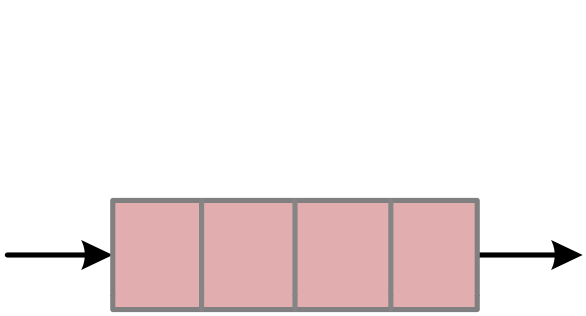
- In this routing arrangement, the nodes are alternately assigned XYand YX for load balancing. The scheme results in balance loads for symmetric traffic.



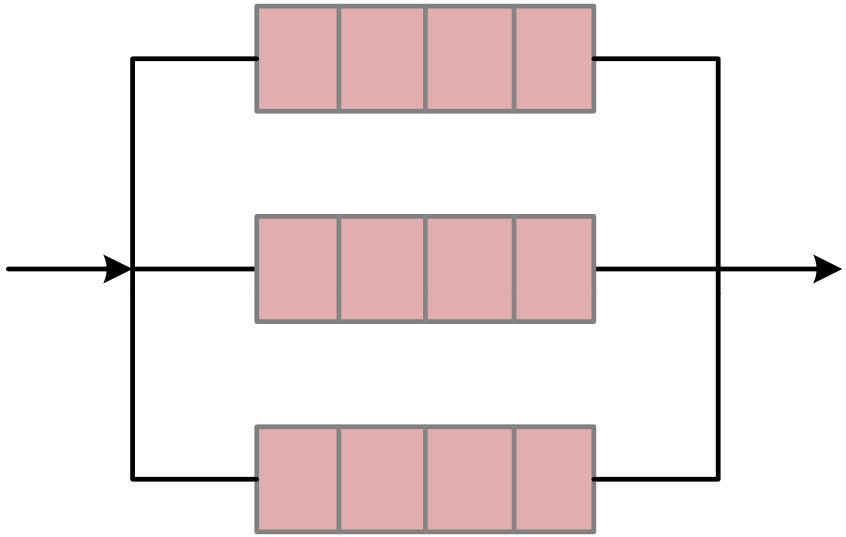
Deflection Routing

- In this routing scheme, packets are normally delivered using predefined shortest-path logic
- In cases where there is contention, with two packets arriving at a router needing to use the same path, the packet with the higher assigned priority gets the allocation and the other packet is deflected to a non-optimal path

Flow Control



(a) One channel per link

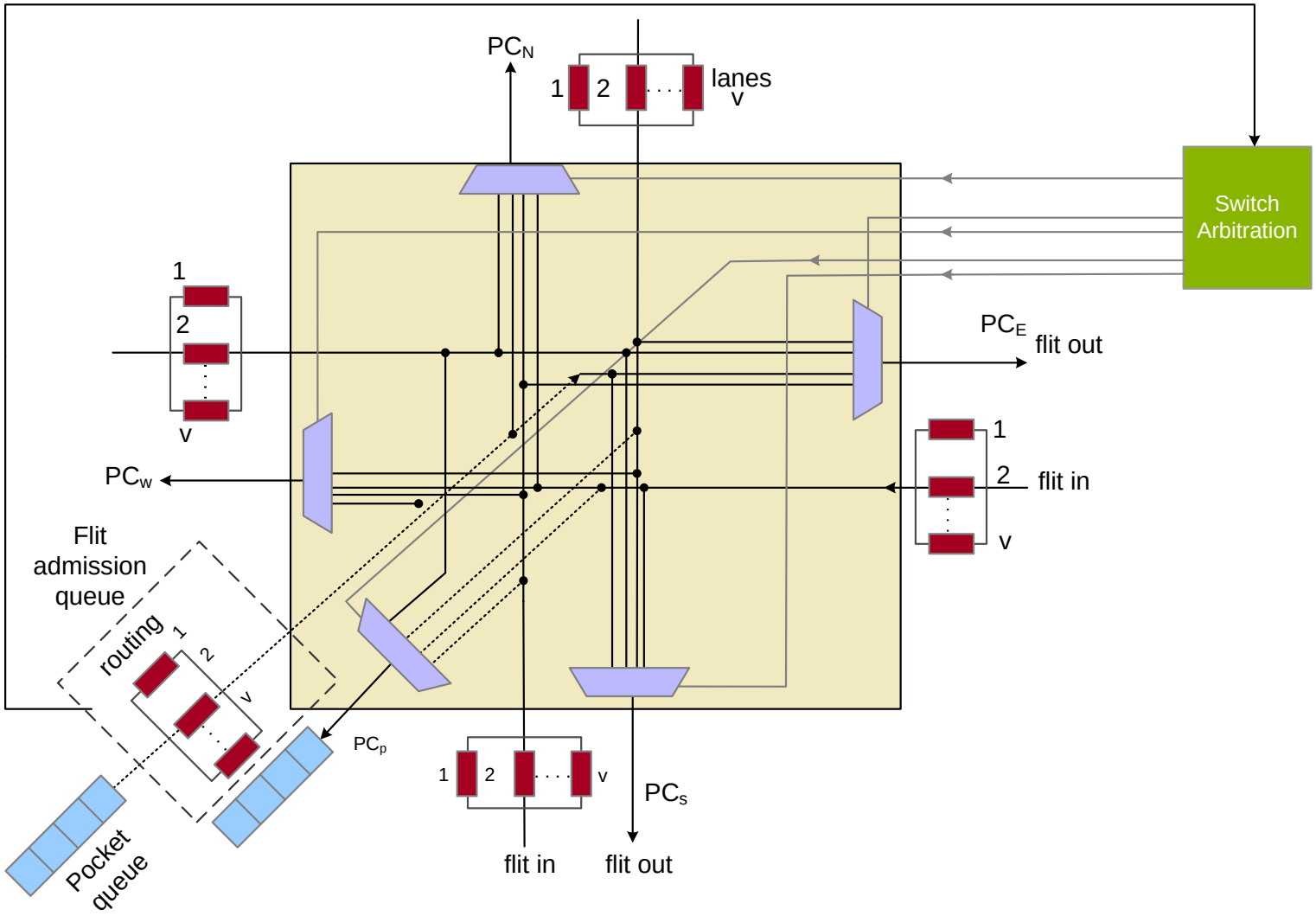


(b) Multiple virtual channels per link

Virtual Channels

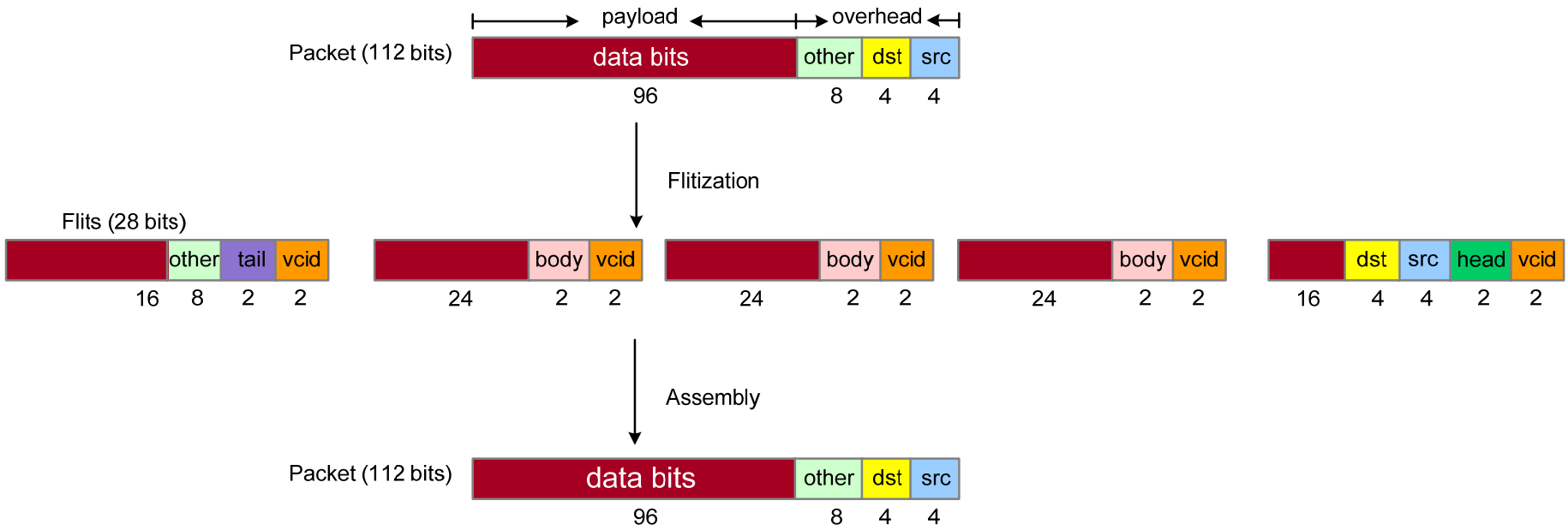
- The addition of virtual channels in NoC architecture provides flexibility but also adds complexity and additional area

Design of a Router for NoC

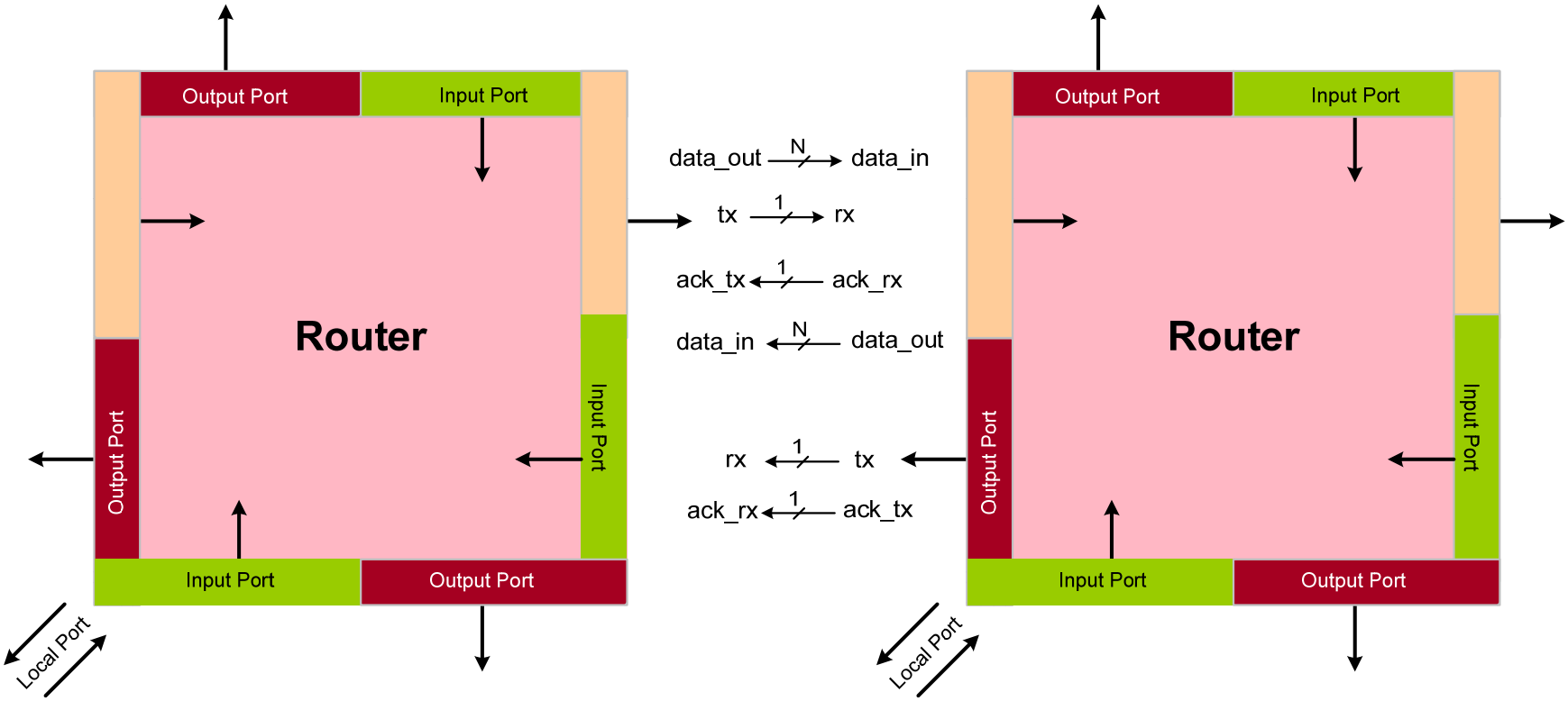


Design of a router with multiple VCs for each physical channel

Process of breaking a packet into a number of flits and assigning a VCid to each



Design of an interface between two adjacent routers



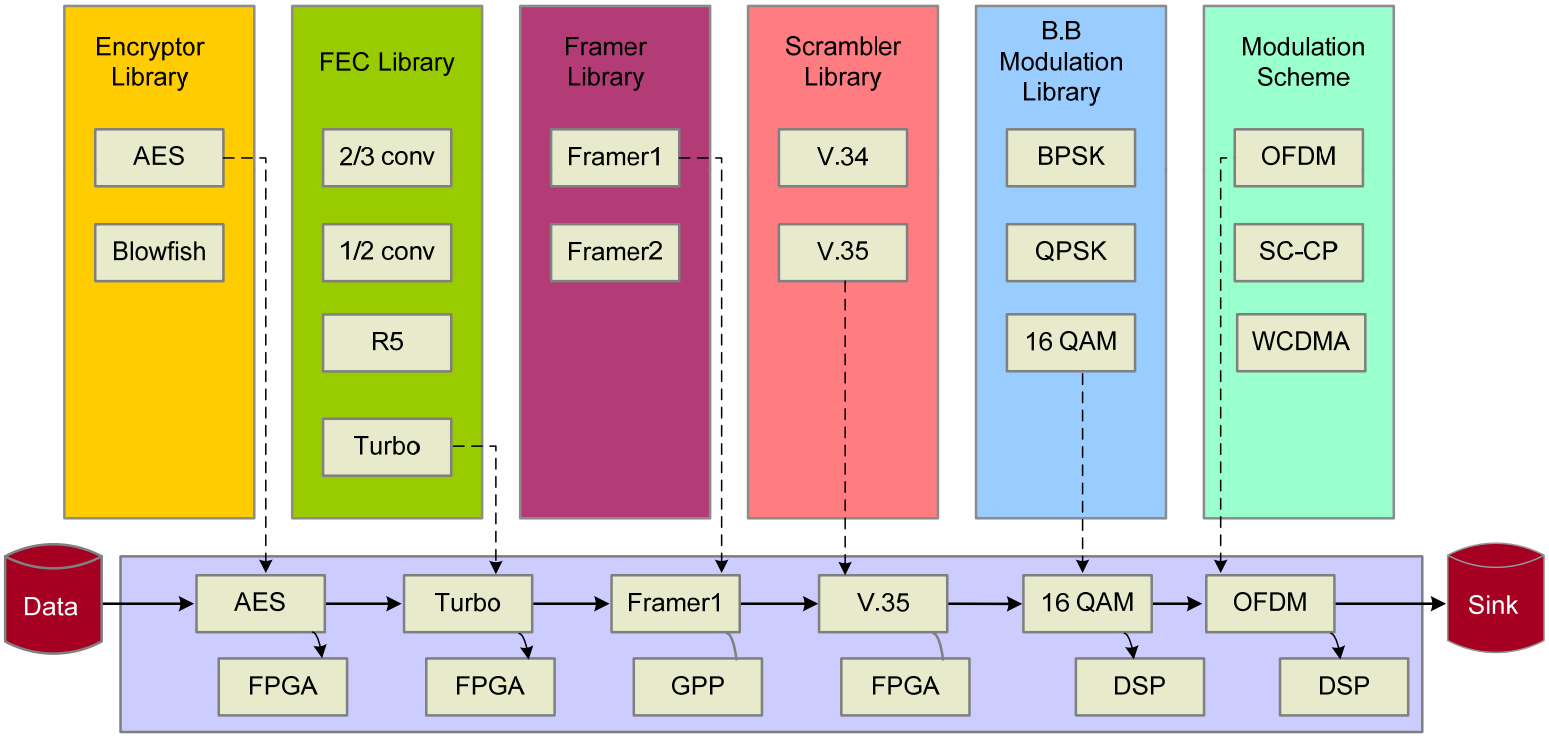
Run-time Reconfiguration

- Many modern digital communication applications need reconfigurability at protocol level
- This requires the same device to perform run-time reconfiguration of the logic to implement an altogether different wireless communication standard
- The device may need to support run-time reconfigurability to implement standards for a universal mobile telecommunications system (UMTS) or a wireless local area network (WLAN)
- The device switches to the most suitable technology if more than one of these wireless signals are present at one location

NoC for Software-defined Radio (SDR)

- SDR involves multiple waveforms. A waveform is an executable application that implements one of the standards or proprietary wireless protocols

  Each waveform consists of a number of components. These components are connected through interfaces with other components. There are standards to provide the requisite abstraction to SDR development

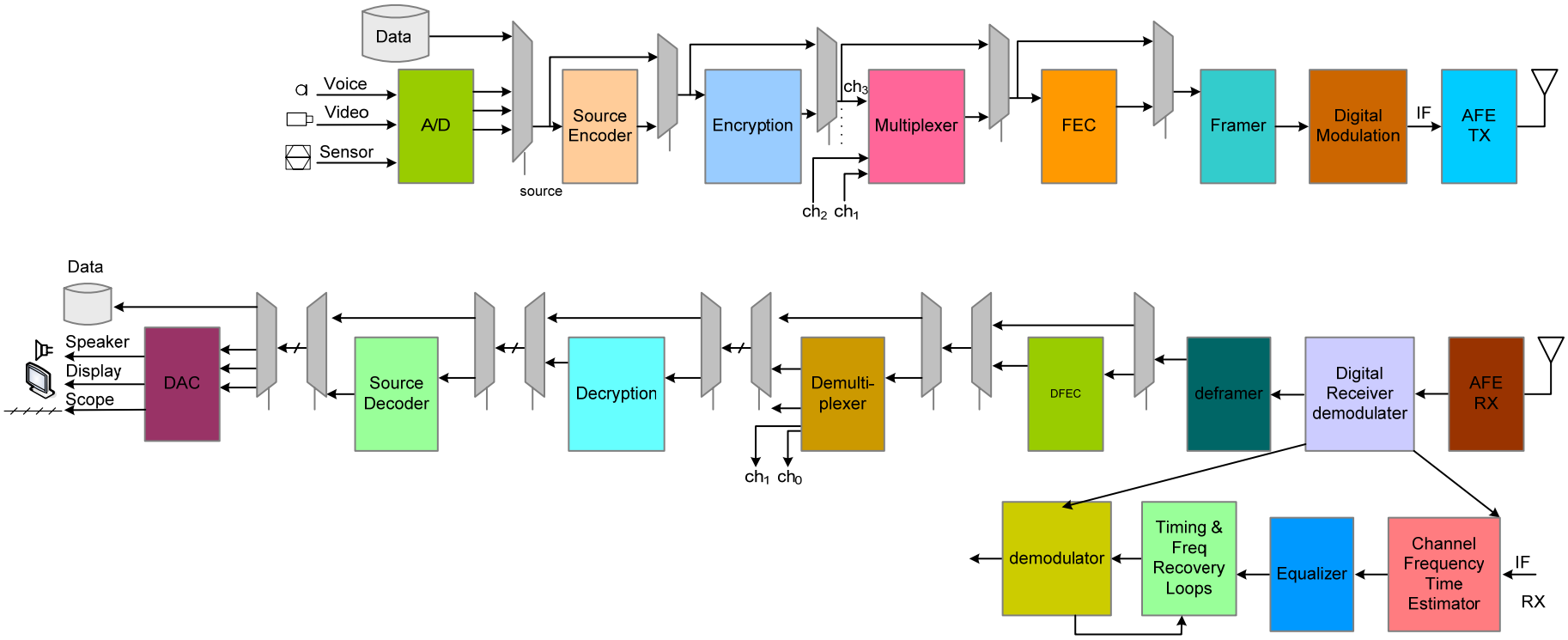


Component-based SDR design. (a) Component library. (b) Waveform creaction and mapping on different target technologies in the system

Typical Digital Communication System

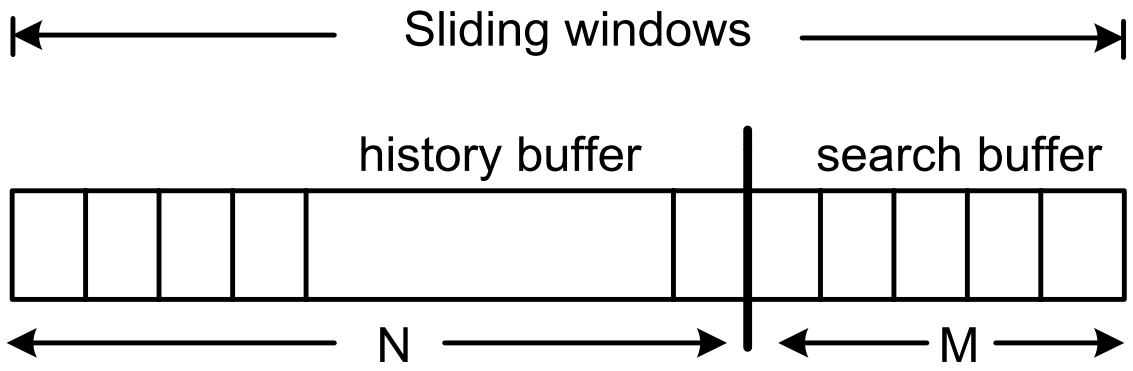
- Source Encoding
- Data Compression
- Encryption

Representative communication system based on digital technologies

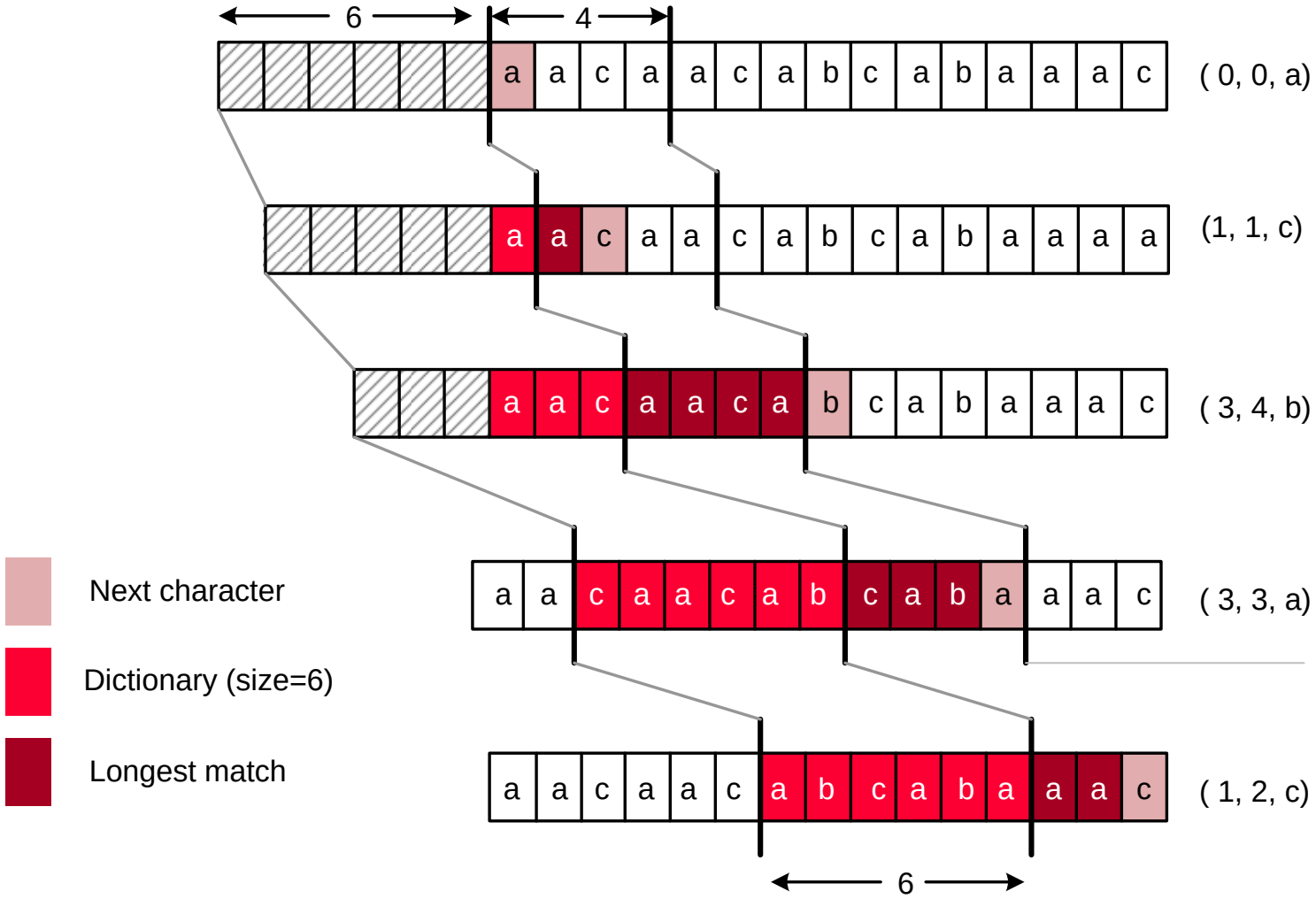


Representative communication system based on digital technologies

Sliding window for data compression algorithm



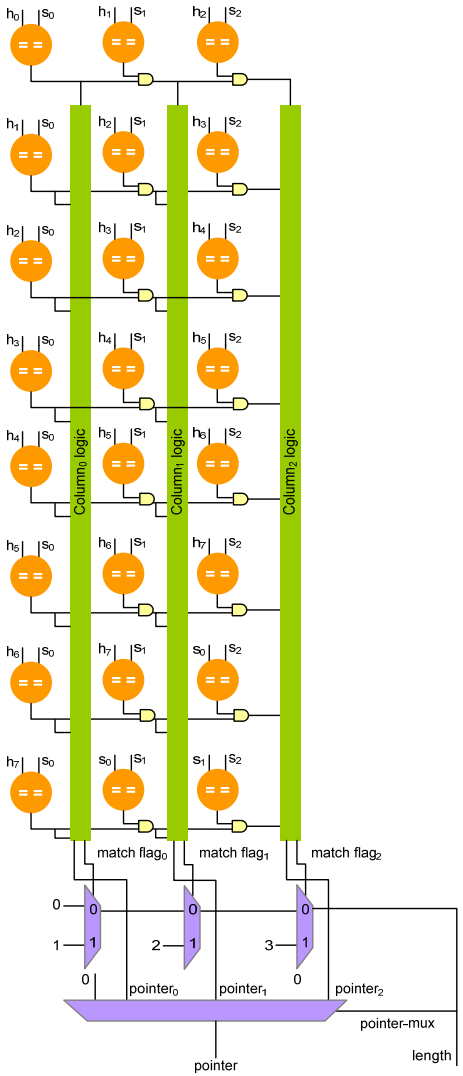
Looking for the longest match



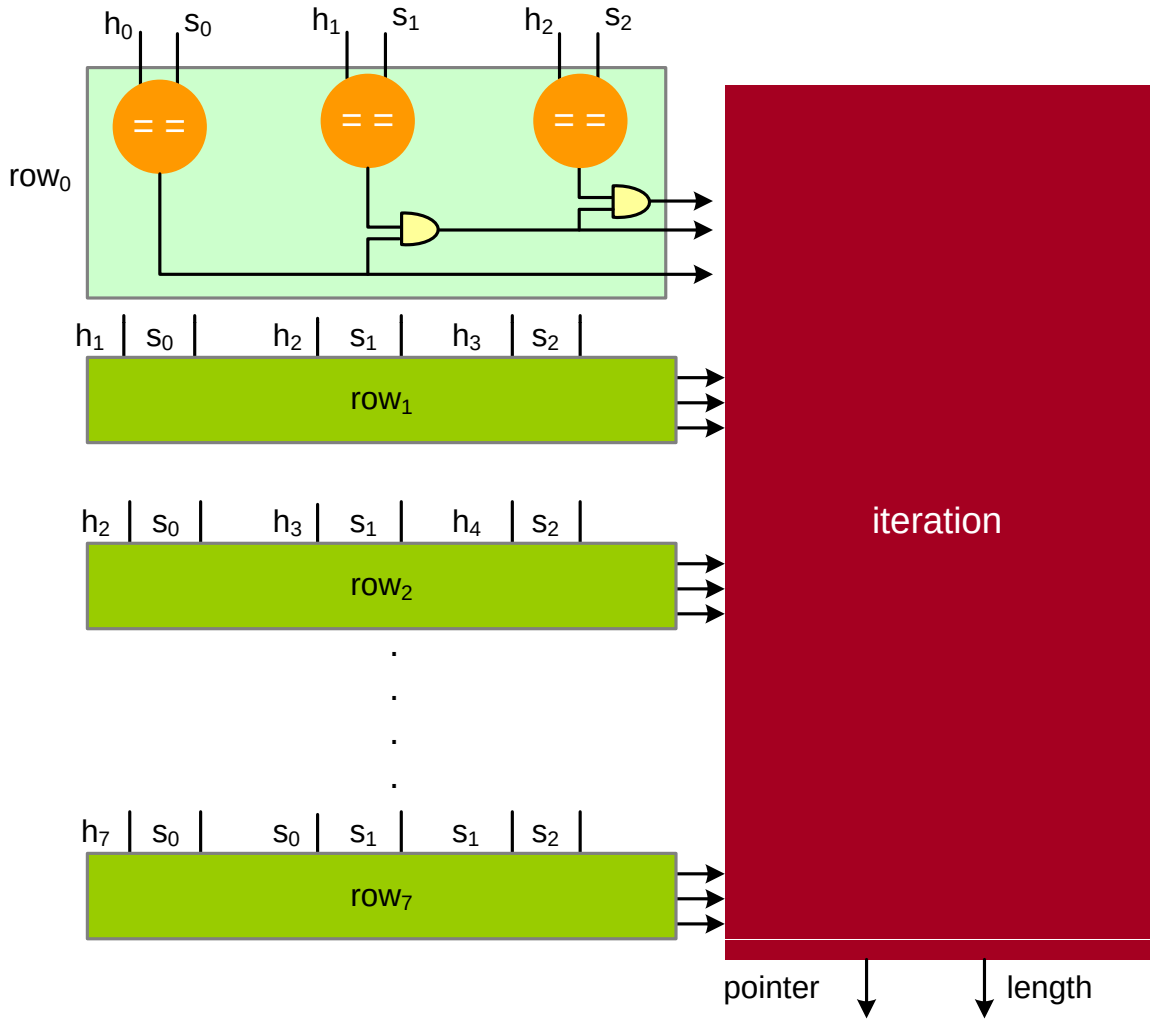
Parallel comparisons of characters listed in first row with the three characters in the search buffer

h0	h1	h2	h3	h4	h5	h6	h7	s0	s1	s2
s0	s1	s2								
	s0	s1	s2							
		s0	s1	s2						
			s0	s1	s2					
				s0	s1	s2				
					s0	s1	s2			
						s0	s1	s2		
							s0	s1	s2	

Digital design for computing iteration of the LZ77 algorithm in parallel. (a) Identifying logic in each column.



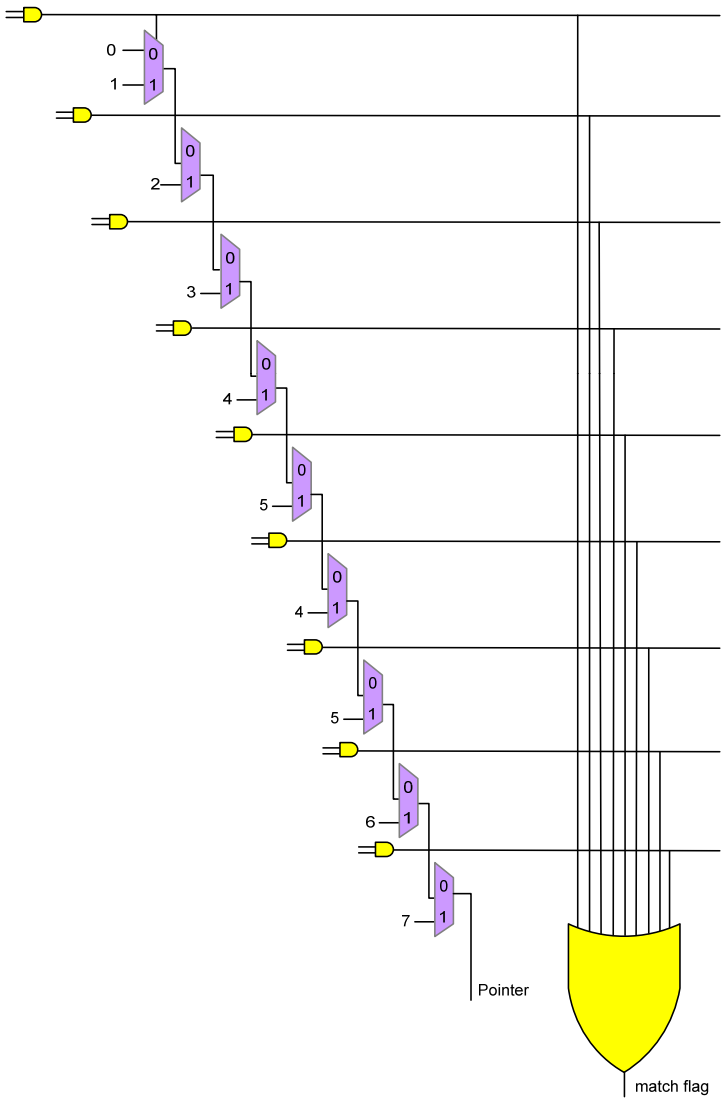
(b) A block representing the logic of three columns and pointer and length calculation



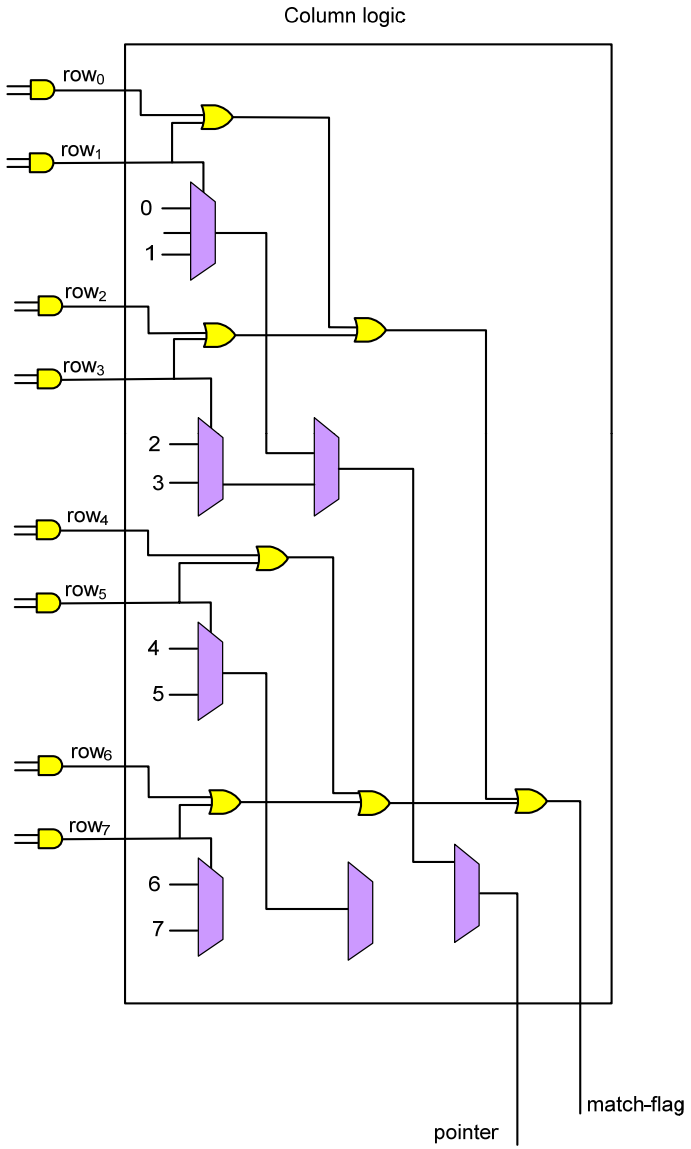
AES Algorithm

- An AES algorithm encrypts 128-bit block of plain text using one of the three different sizes of keys
- The key sizes are 128, 192, and 256. The encryption is performed in multiple rounds.

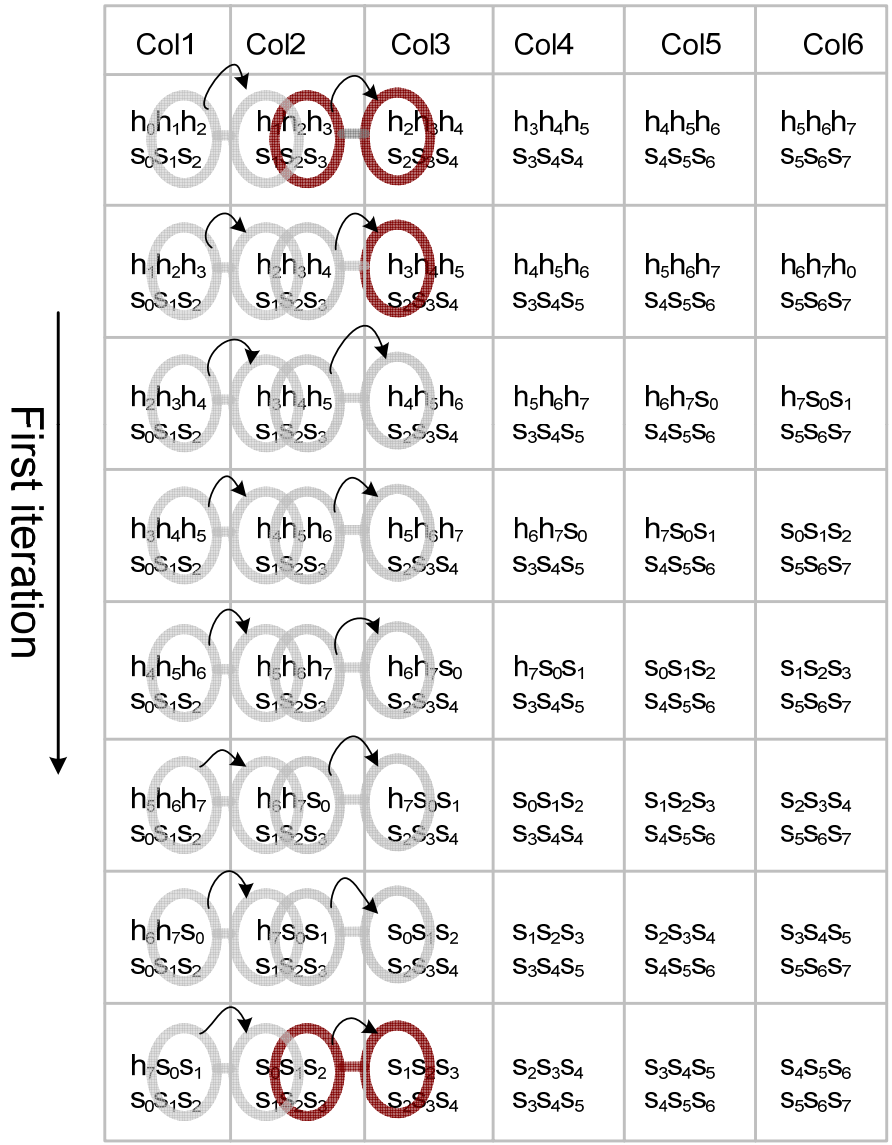
Pointer calculation logic for each column: (a) with serial complexity, and (b) with logarithmic complexity



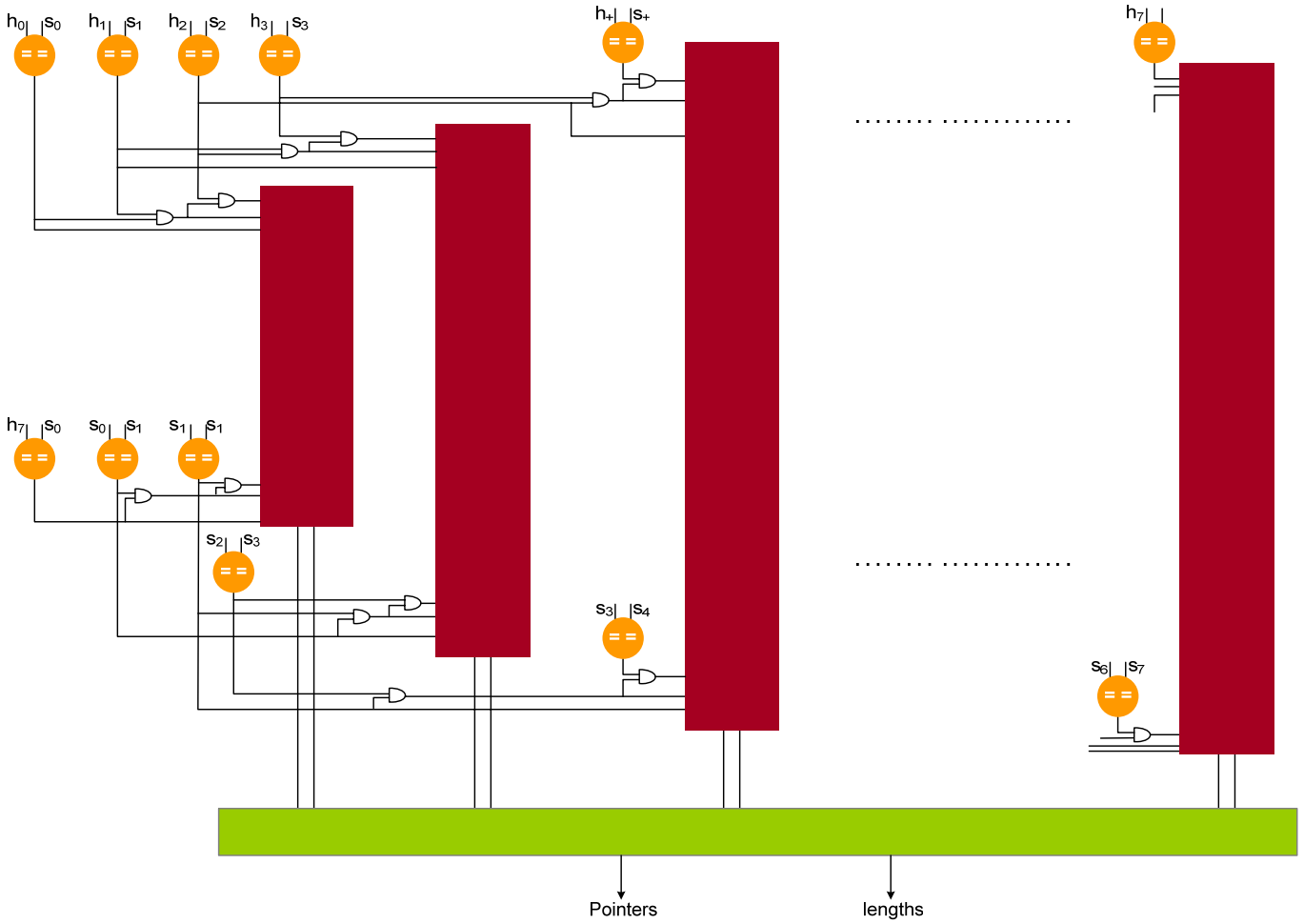
(b) with logarithmic complexity



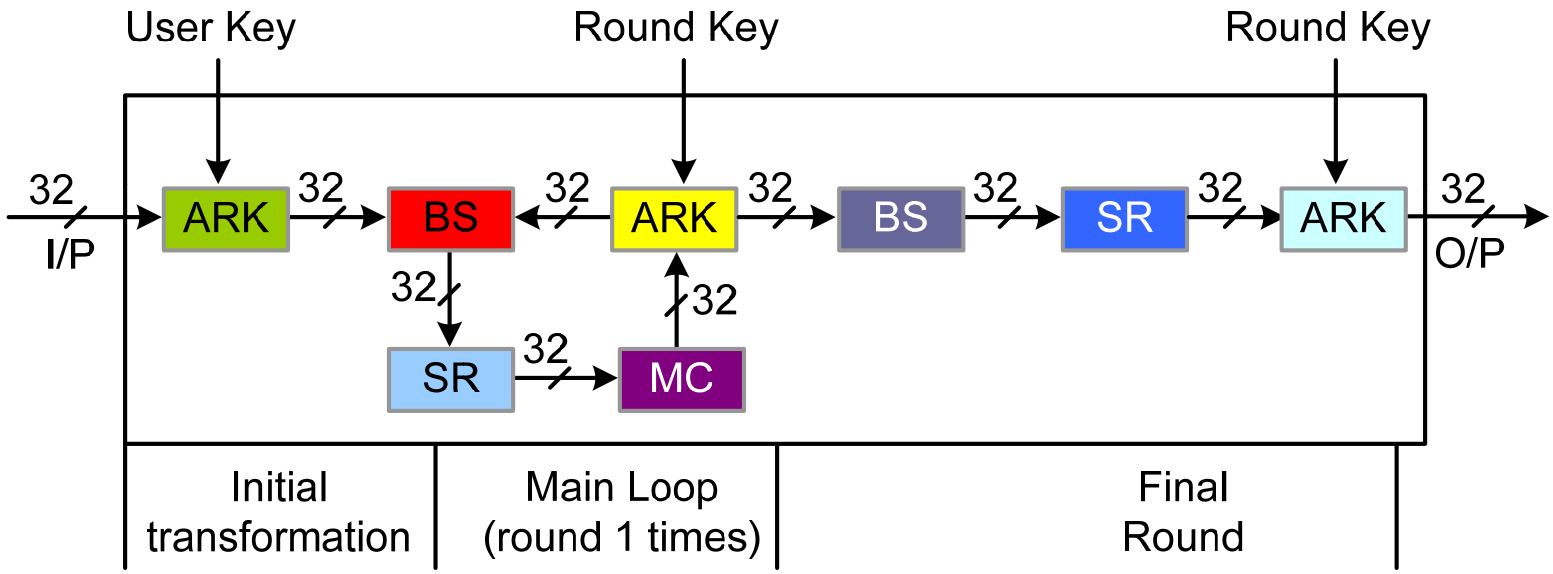
Multiple iteration of the algorithm with shared comparisons



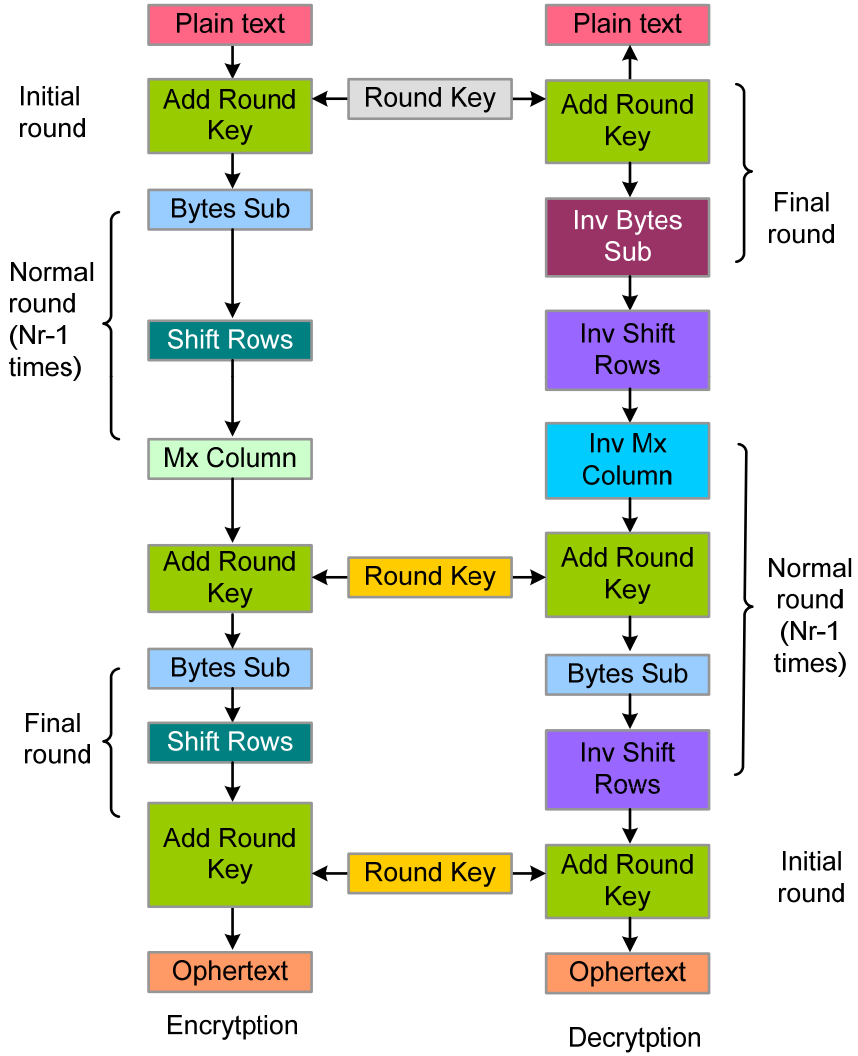
Digital design implementing multiple iterations of the LZ77 algorithm



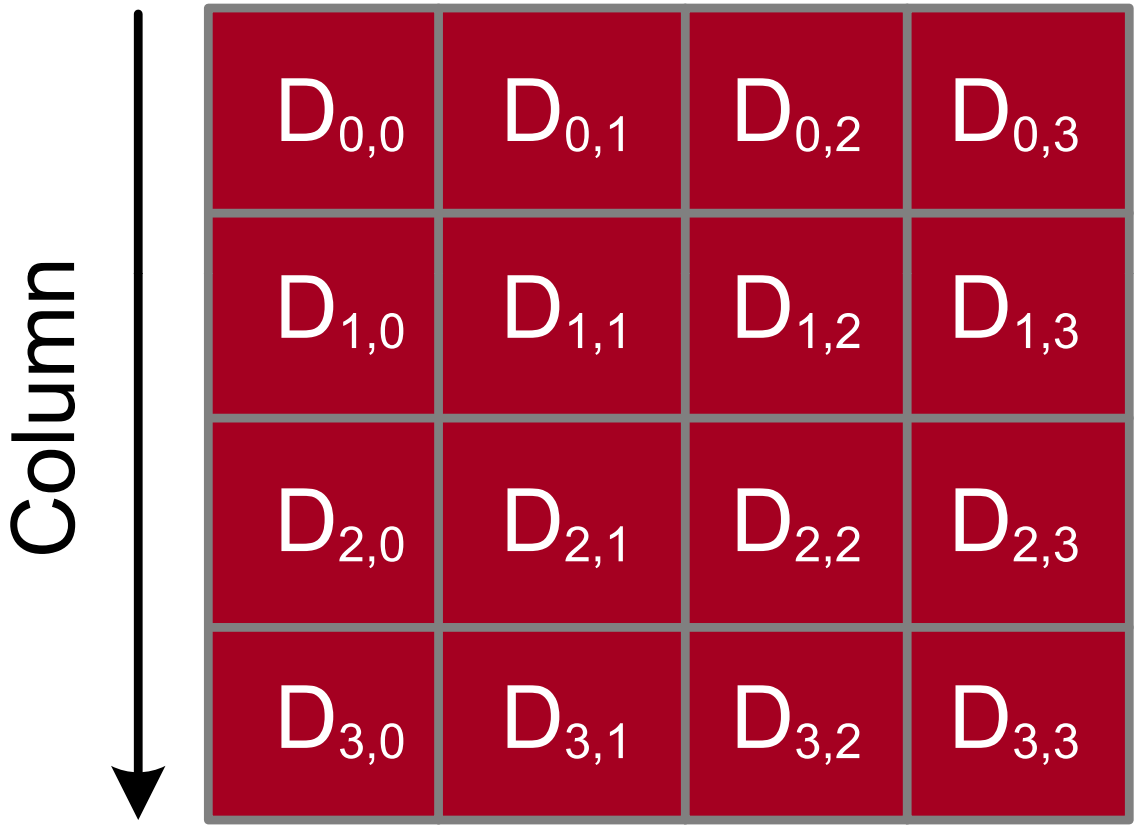
AES algorithm (a) Block diagram of encryption. (b) Flowof algorithm for encryption and decryption



(b) Flow of algorithm for encryption and decryption

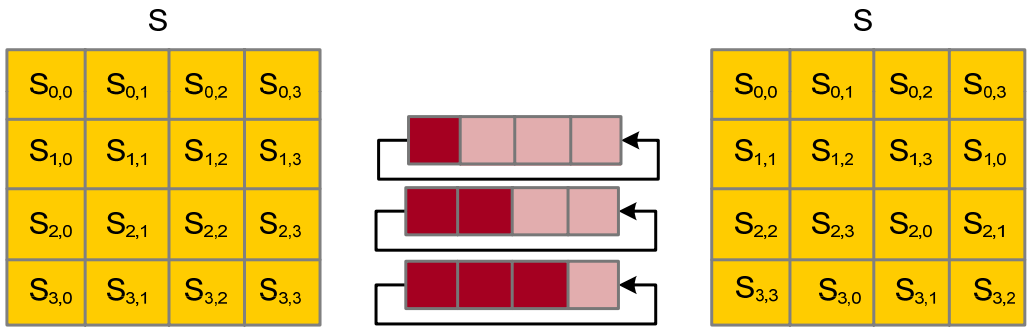


Filling of plain text in a state matrix



(a) Shift row operation of AES algorithm

(b) Matrix multiplication of mixed column operation



(a)

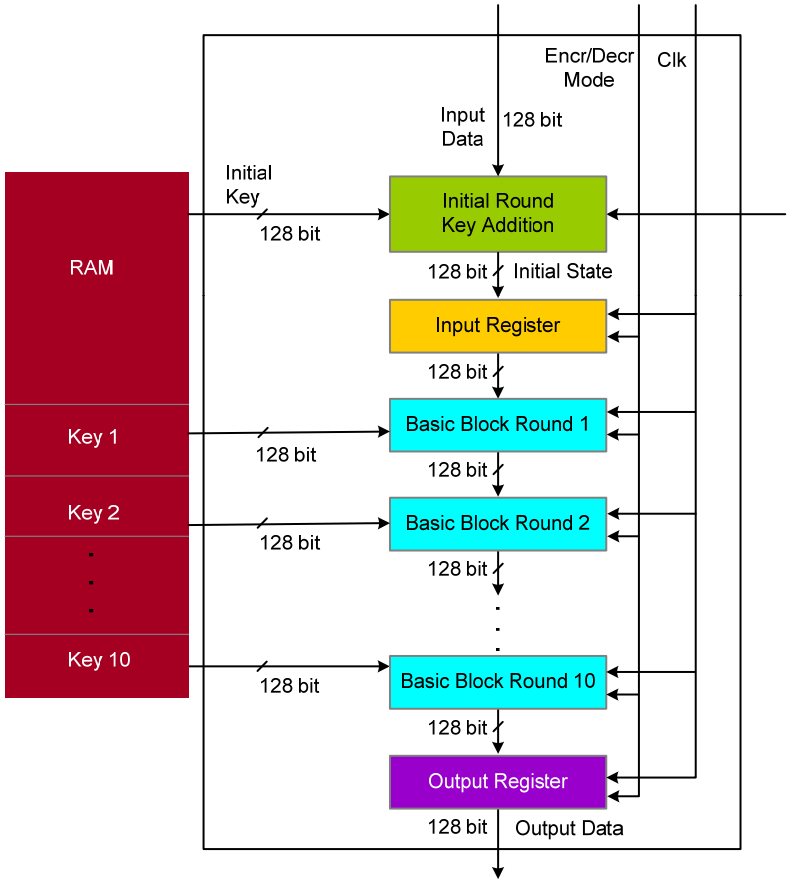
$$\begin{bmatrix} C'_{0,j} \\ C'_{1,j} \\ C'_{2,j} \\ C'_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} C'_{0,j} \\ C'_{1,j} \\ C'_{2,j} \\ C'_{3,j} \end{bmatrix}$$

(b)

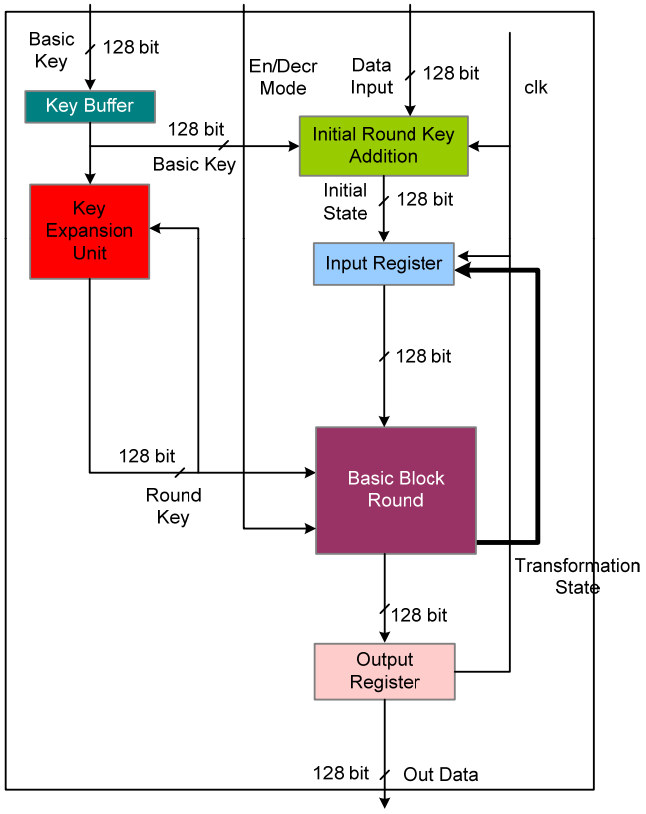
AES Architectures

- Selection of a particular implementation of the AES algorithm primarily depends on the input data rate
- A pipelined fully parallel architecture implements all the iterations in parallel
- There are many applications where the architecture is needed to support moderately high data rates in a small area, as is the case with most of the wireless communication standards

(a) Pipelined fully parallel design of AES (b) Time shared design

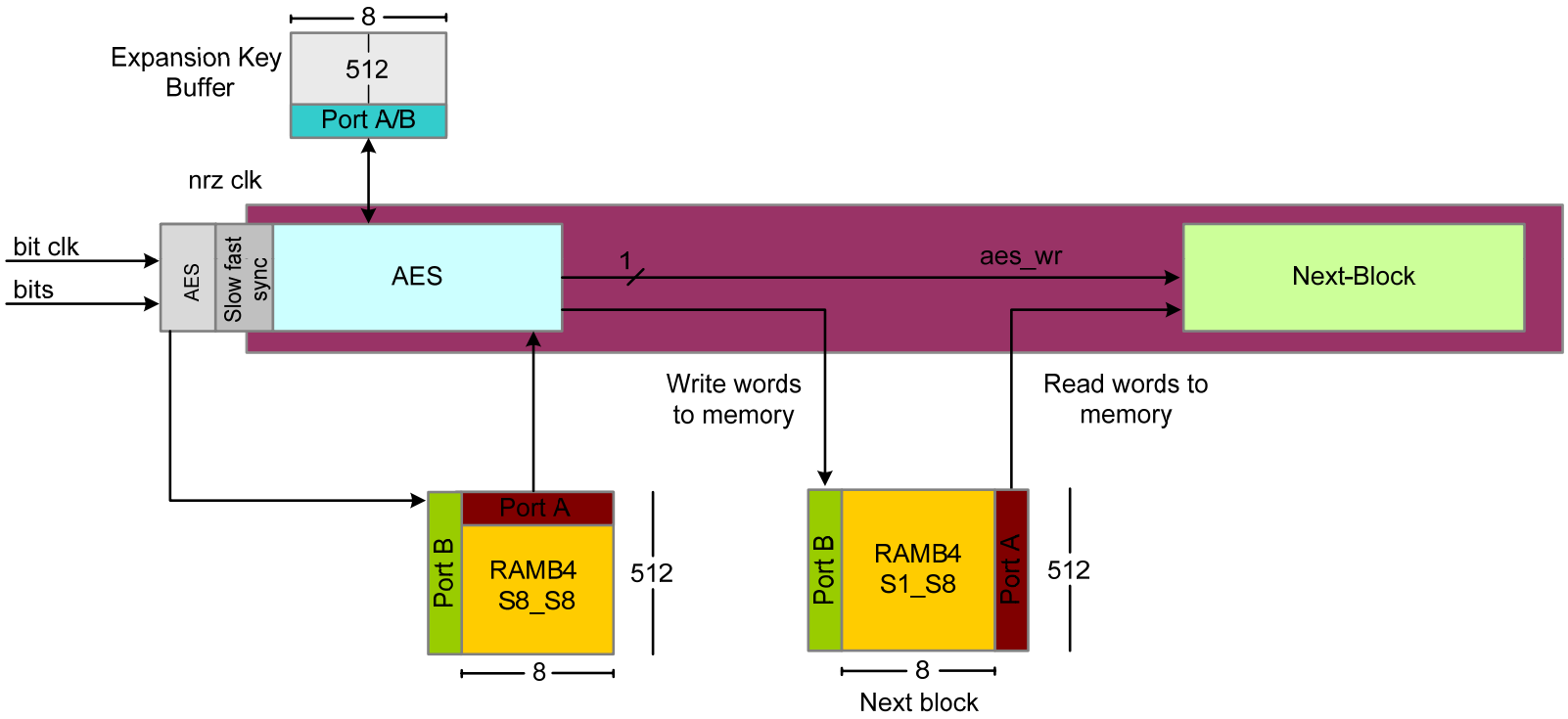


(a)

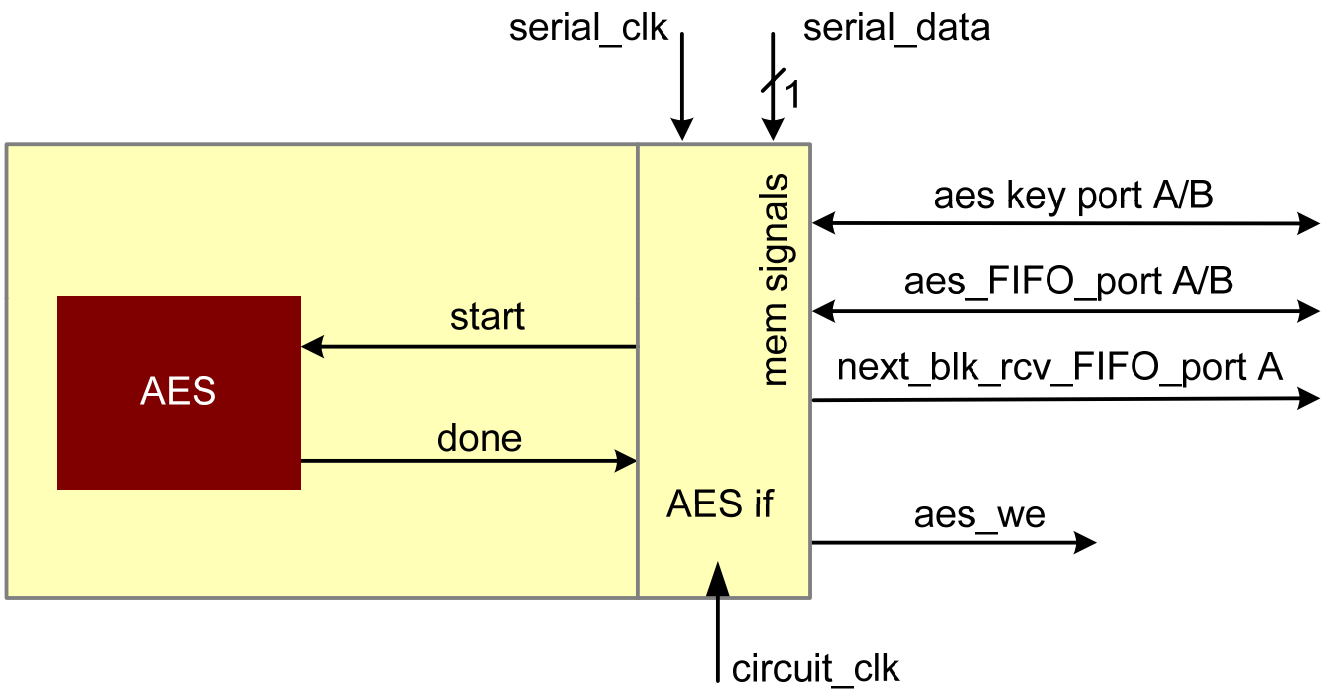


(b)

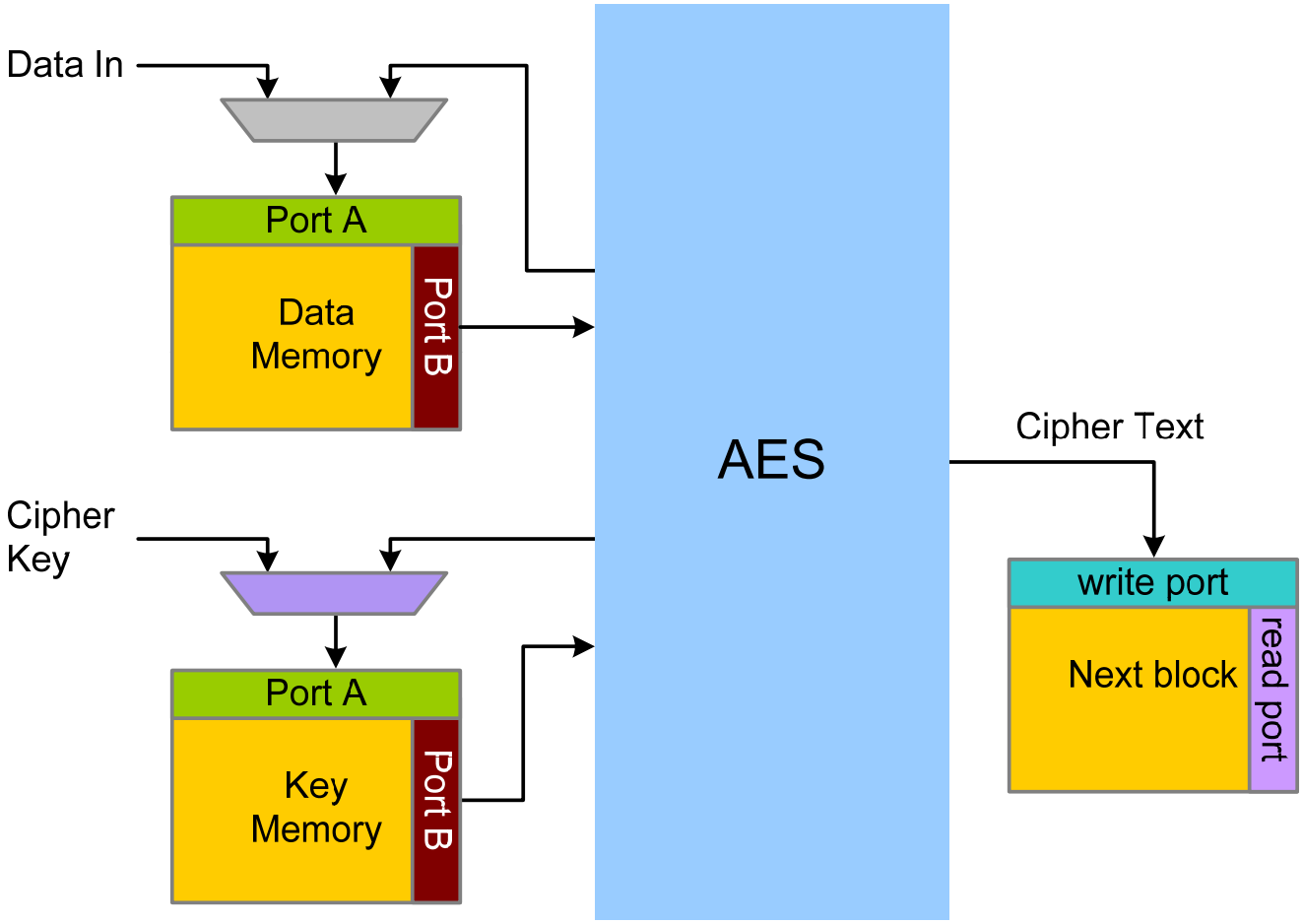
KPN-based top-level design incorporating an AES processor



Top-level design of AES engine



Memory configuration of AES engine in KPN settings



Contd…

- There are two types of memory in most of the FPGAs:
  - distributed RAM and
  - block RAM
- The LUTs can be configured as distributed RAM whereas block RAM is a dedicated dual-port memory.
- An FPGA contains several of these blocks

Time-Shared 8-bit Folding Architecture

- No straightforward method of further folding AES architecture as the algorithm is iterative and nonlinear while it performs computation for a round

$$addrSR = (addrSR + 5)\%16$$

$$addrKey = \{roundCount, addrSM\}$$

$$\begin{bmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{03} \end{bmatrix} = \begin{bmatrix} 02\ 03\ 02\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} C_{00} \\ C_{01} \\ C_{02} \\ C_{03} \end{bmatrix}$$

State indexing for shift-row operation: (a) original indexing, and (b) shift-row indexing

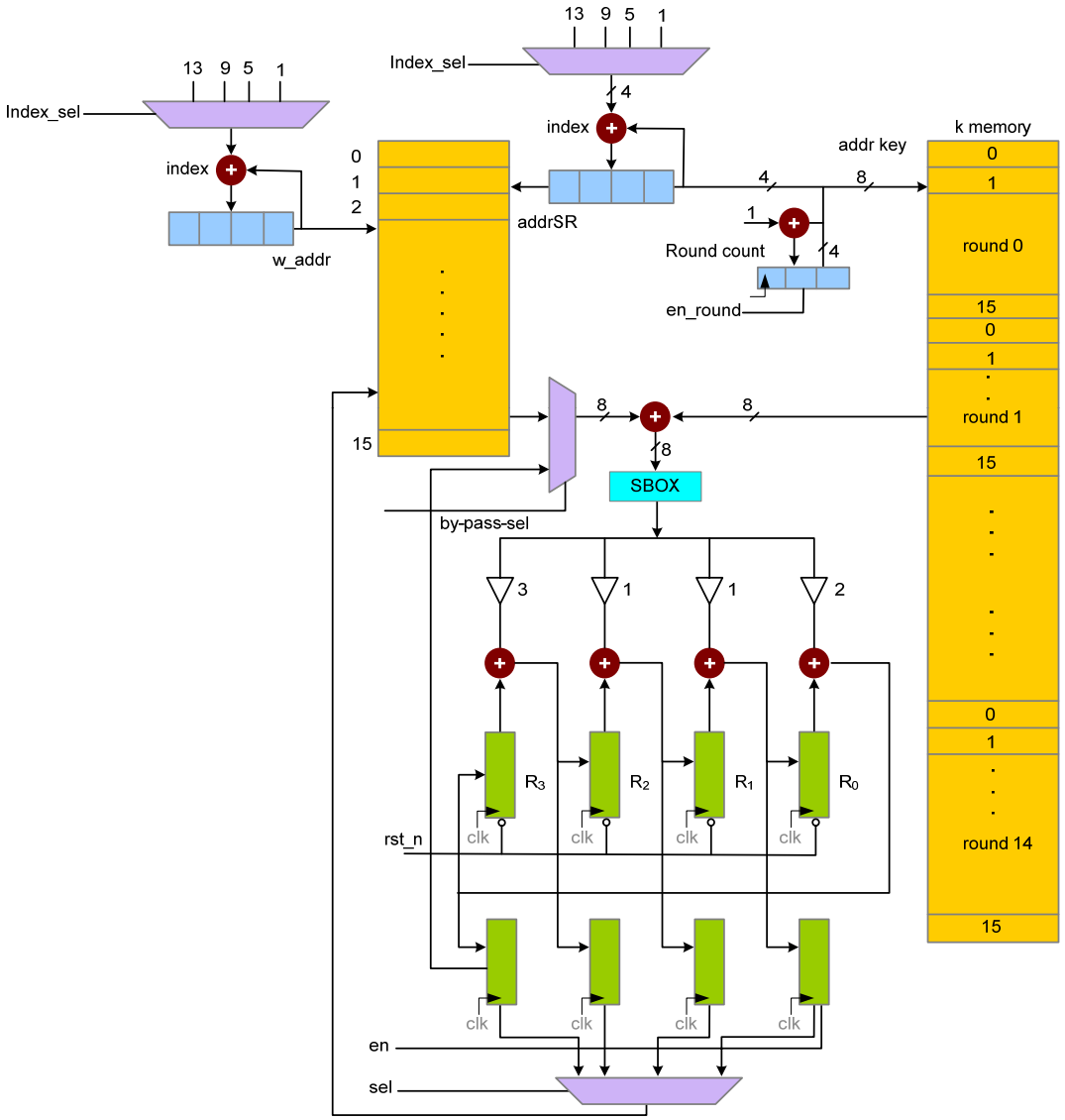
(0,0) 0	(0,1) 4	(0,2) 8	(0,3) 12
(1,0) 1	(1,1) 5	(1,2) 9	(1,3) 13
(2,0) 2	(2,1) 6	(2,2) 10	(2,3) 14
(3,0) 3	(3,1) 7	(3,2) 11	(3,3) 15

(a)

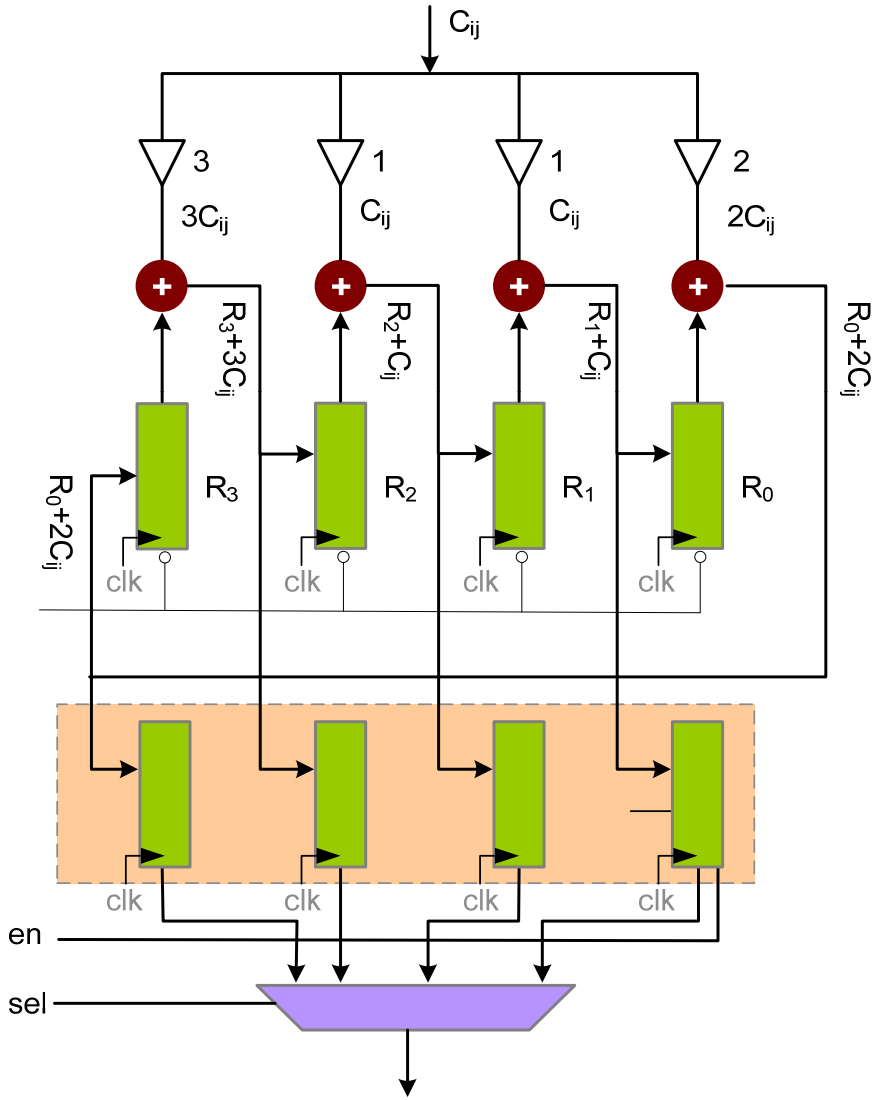
(0,0) 0	(0,1) 4	(0,2) 8	(0,3) 12
(1,1) 5	(1,2) 9	(1,3) 13	(1,0) 1
(2,2) 10	(2,3) 14	(2,0) 2	(2,1) 6
(3,3) 15	(3,0) 3	(3,1) 7	(3,2) 11

(b)

Eight-bit time-shared AES architecture



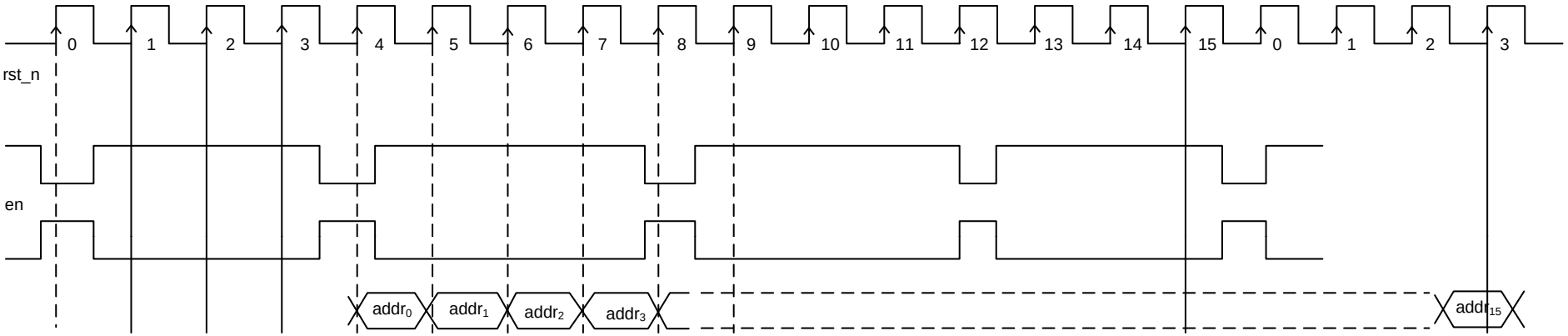
Mix-column design for byte systolic operation



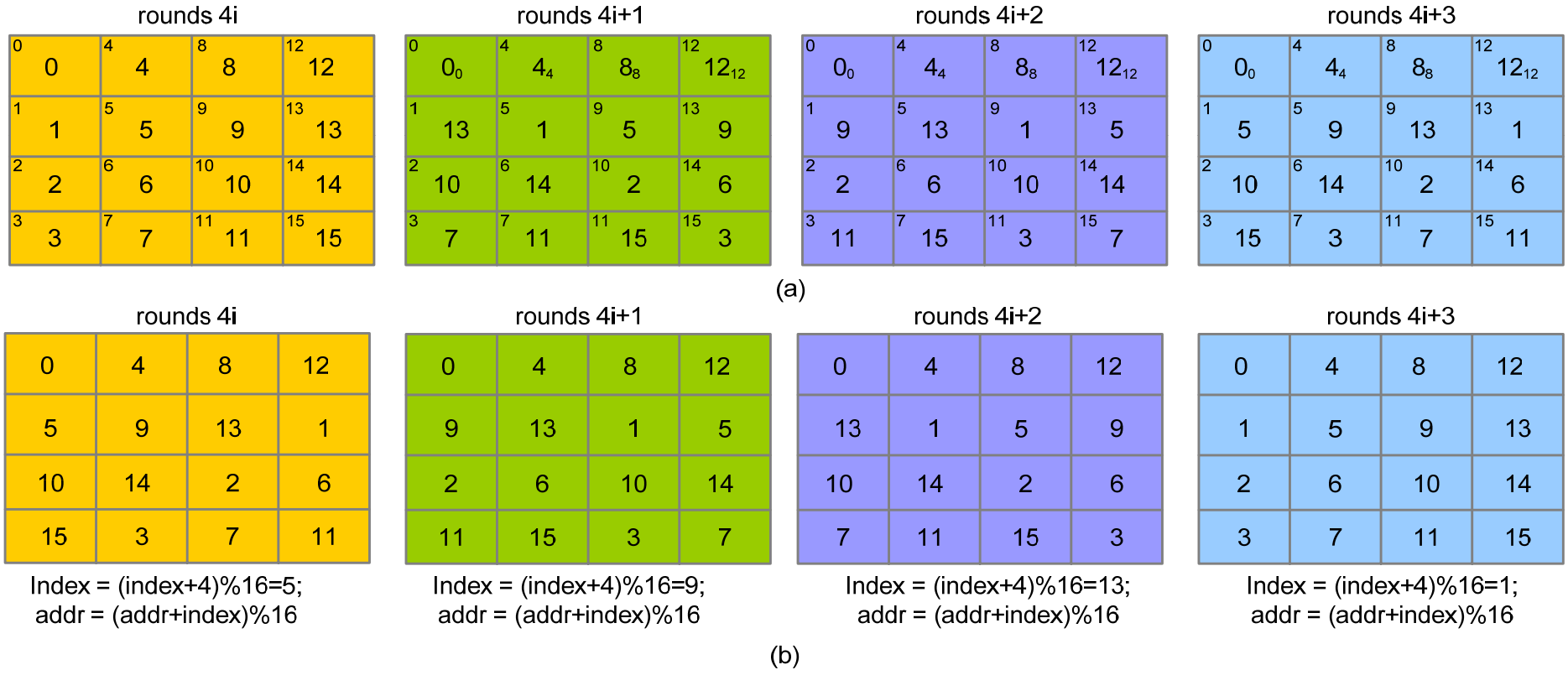
Byte-Systolic Fully Parallel Architecture

- Works on byte in-place indexing
- A byte of plain text is input to the architecture and a byte of cipher text is output in every clock cycle after an initial latency of 1615 cycles
- All the rounds of encryption are implemented by cascading all the stages with pipeline logic
- The data is input to the first stage in byte-serial fashion
- When the 16 bytes have been written in the first dataRAMblock, the stage starts executing the first round of the algorithm. At the same time the input data for the second frame is written in the RAM block by employing byte in-place addressing
- This scheme writes the input data at locations that are already used in the current cycle of the design.

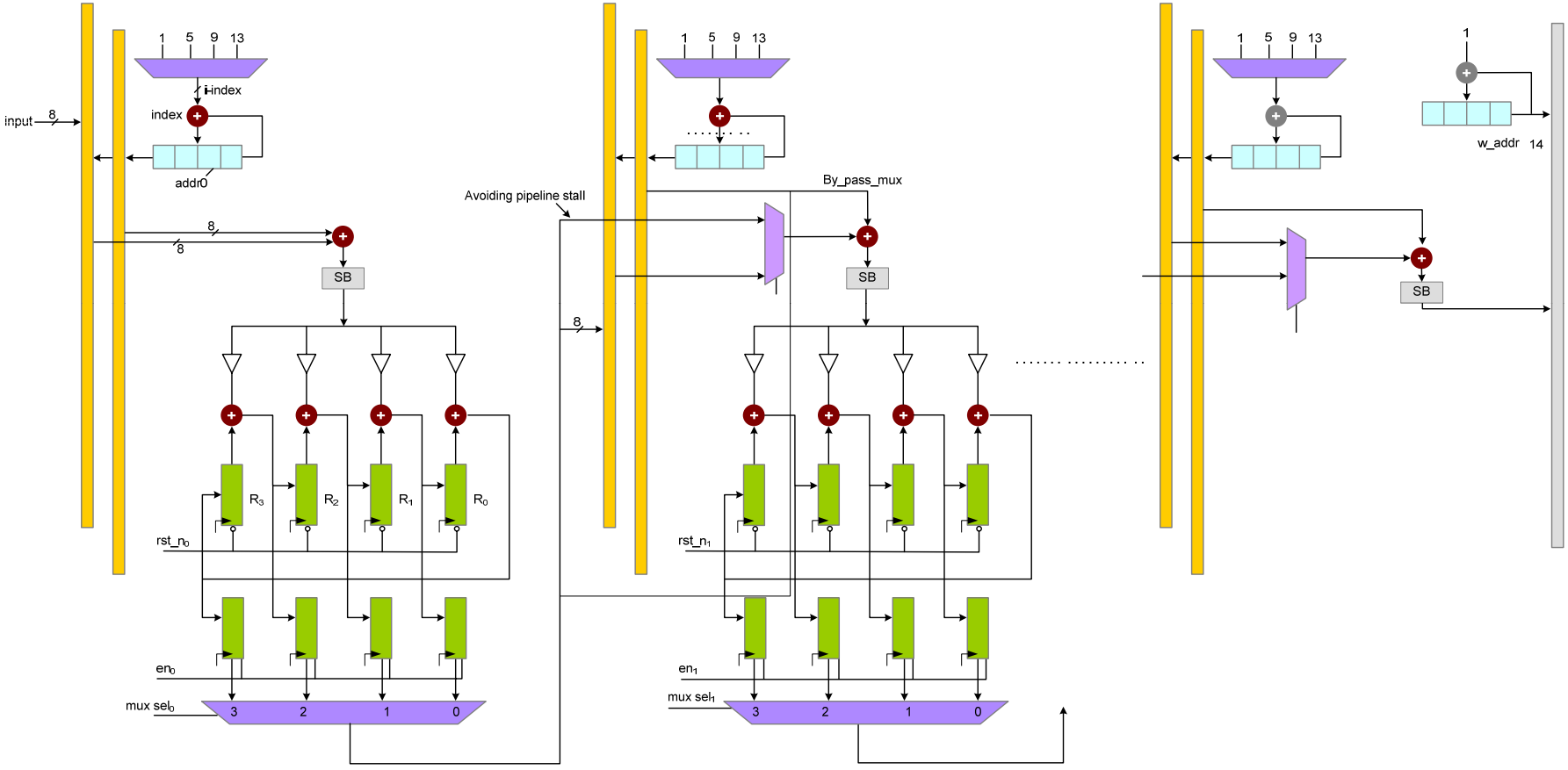
Timing diagram for the time-shared AES architecture



Byte in-place indexing for byte systolic AES architecture. (a) Indices for writing data for first four frames. (b) Memory locations for reading data in shift-row format



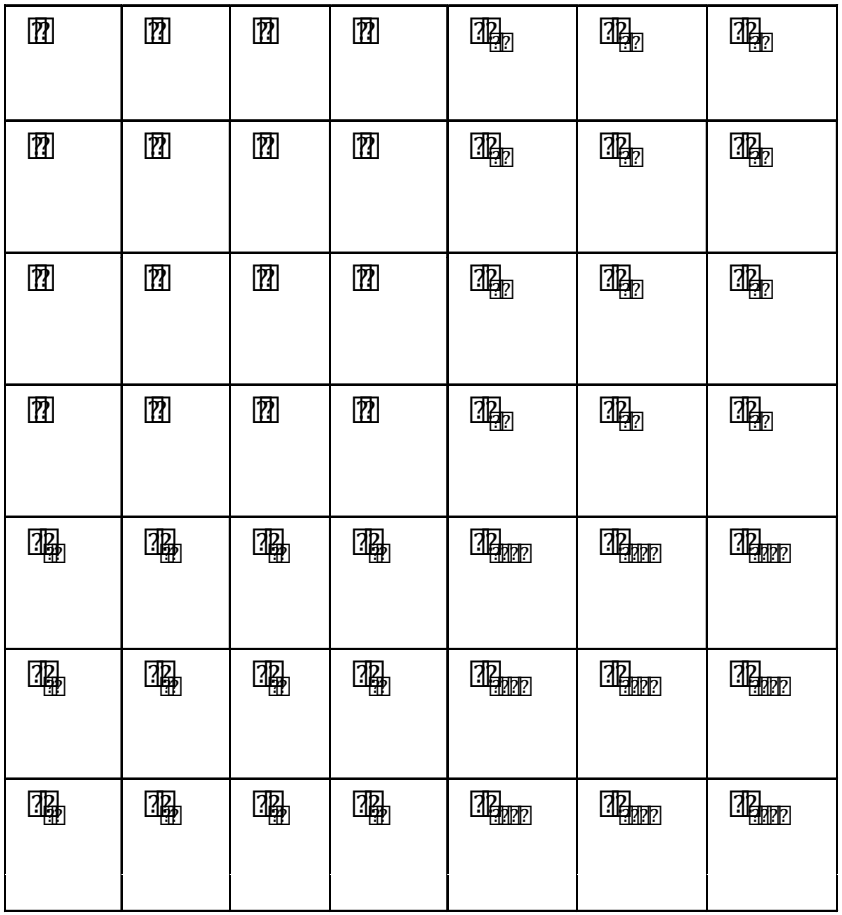
Byte systolic architecture implementing AES algorithm



Channel Coding

- In many communication systems, the next block in a transmitter is channel coding. This is performed at the transmitter to automatically correct errors at the receiver
- The transmitter performs coding of the input data using convolution, Reed–Solomon or turbo codes
- In many applications more than one type of error-correction coding is also performed
- All these algorithms can be mapped in HW for high-throughput applications

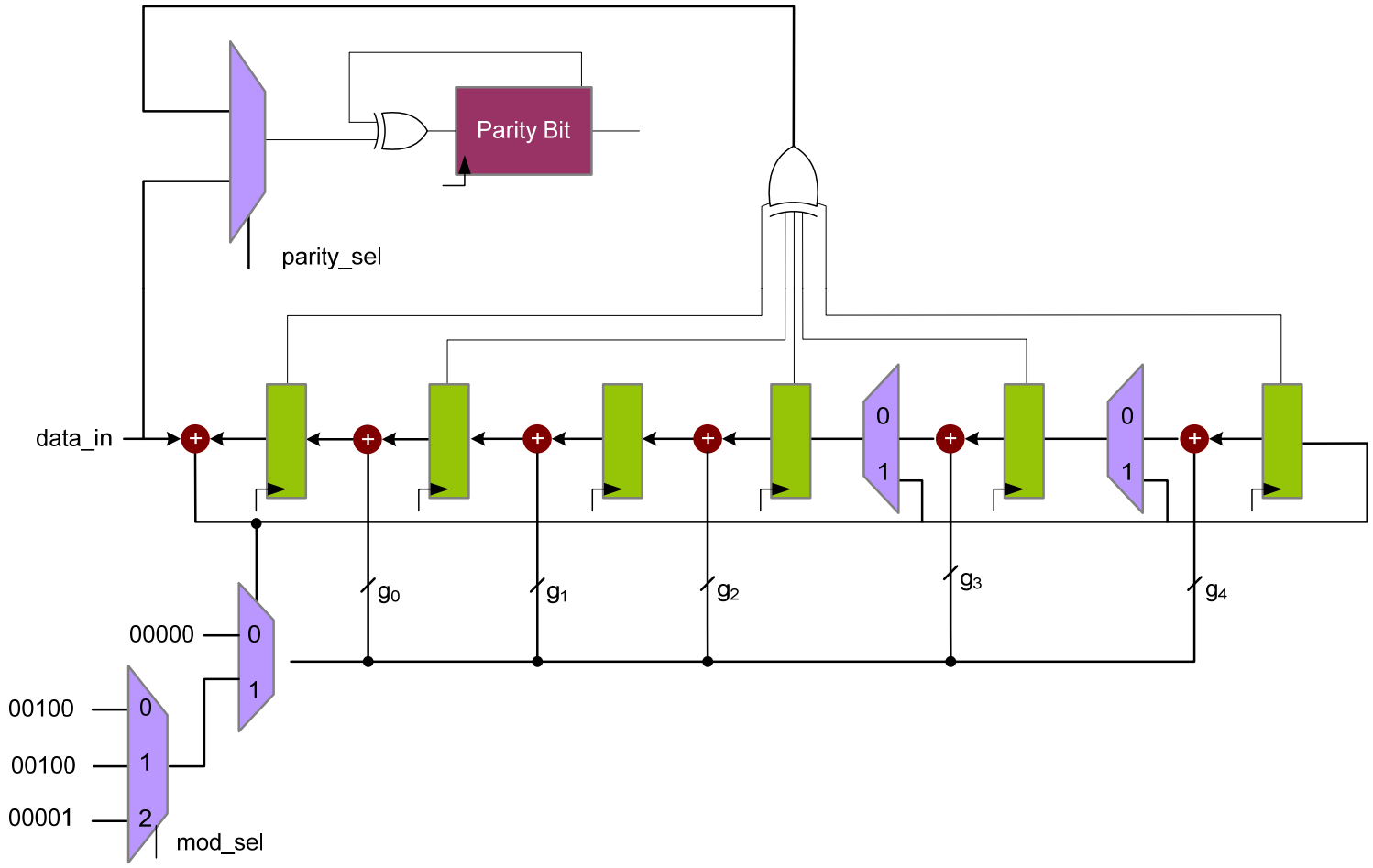
(49,16) block turbo code formulation by combining (7,4)(7,4) codes



Options for block turbo code for supporting different code rates

Mode number	Input block dimensions		Input block size	Output block dimensions		Output block size	Code rate
	Rows	Columns	K	Rows	Columns	n	k/n
4	11	11	121	16	16	256	0.47
1	11	26	286	16	32	512	0.55
2	11	57	627	16	64	1024	0.61
9	26	11	286	32	16	512	0.55
5	26	26	676	32	32	1024	0.66
3	26	57	1482	32	64	2048	0.72
10	57	11	627	64	16	1024	0.61
11	57	26	1482	64	32	2048	0.72
6	57	57	3249	64	64	4096	0.79

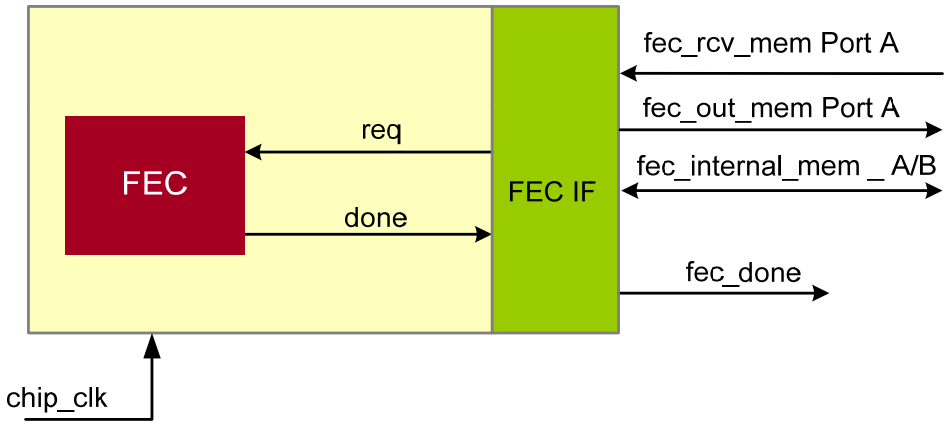
Standard Hamming encoder for BTC



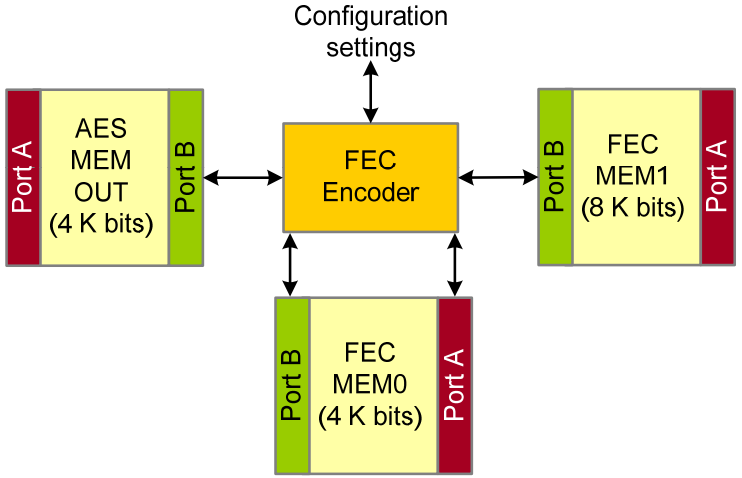
Framing

- For synchronization, different delimiters are inserted in data
- For example, to indentify and synchronize an AES frame, an 8-bit AES header can be inserted after every defined number of AES buffers
- Similarly, for synchronizing an FEC block, a header may also be inserted for synchronization

BTC system-level design (a) BTC interfaces. (b) BTC memory block settings



(a)

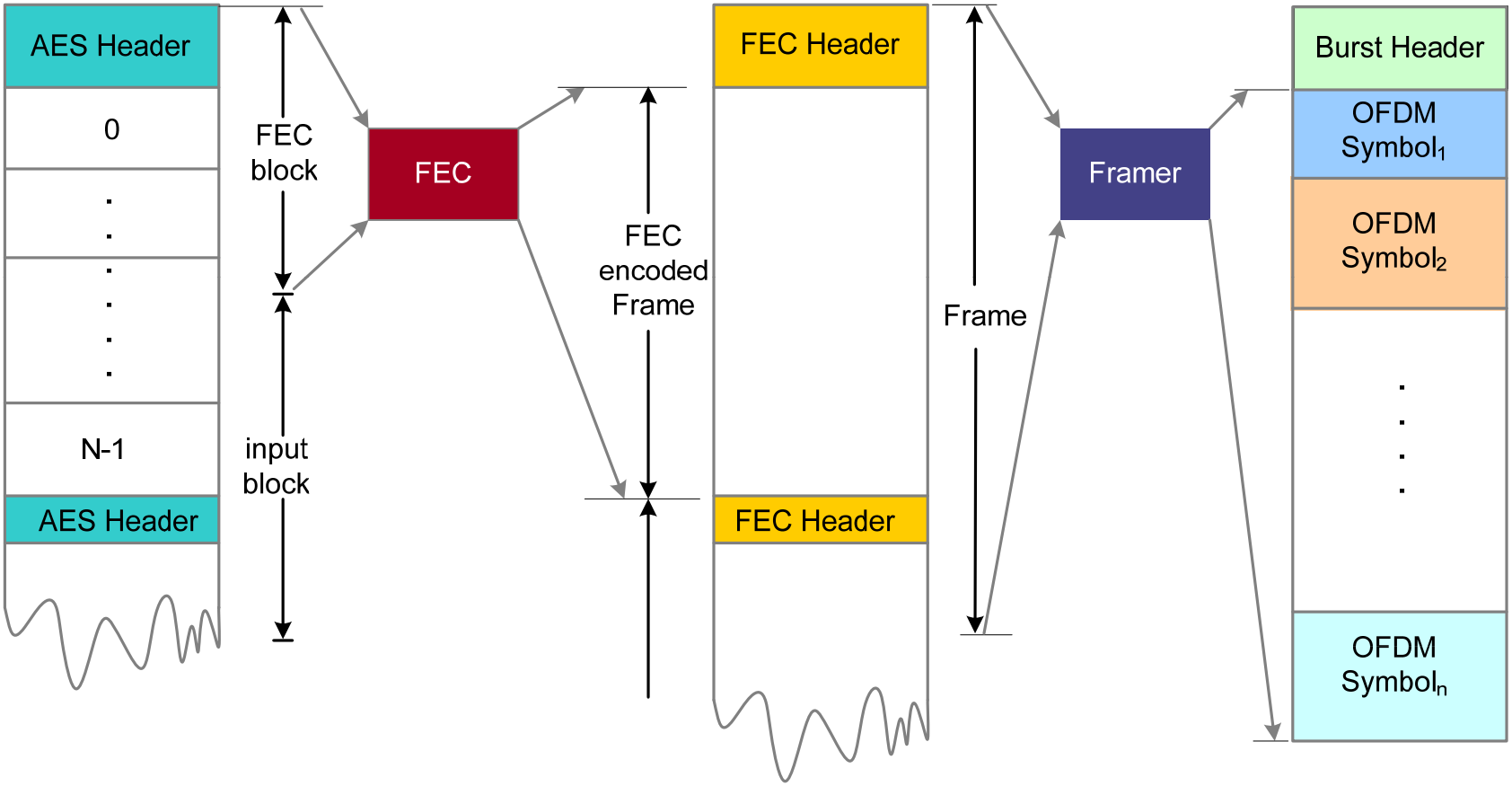


(b)

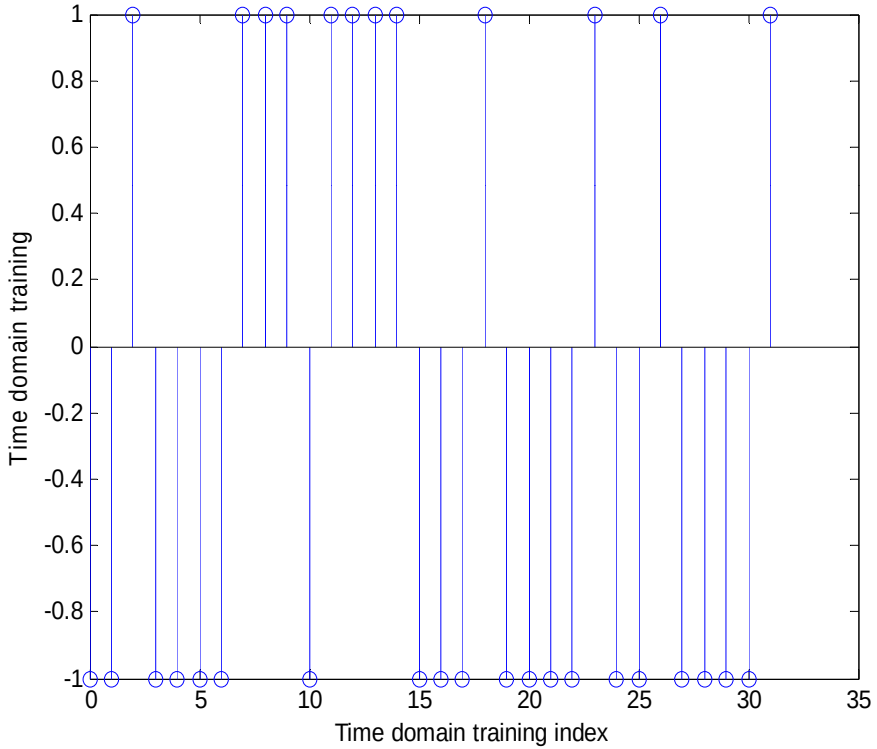
Modulation

- Digital Basenand Modulation
  - Modulation is performed next in a digital communication system.
  - A host of modulation techniques are available, which may be linear or nonlinear
  - They may use single or multiple carriers.
  - They may also perform time-, frequency- or code-division multiplexing for multiple users

Header and frame insertion for synchronization



Time domain training sequence



OFDM-based Digital Transmitter and Receiver

- Owing to its robustness for multiple path fading, it is very effective for high-rate mobile wireless terminals
- The scheme is used in many modern communication system standards for wireless networks, such as IEEE 802.11(a) and 802.16(a), and digital broadcasting such as DAB, DVB-T and DRAM
- OFDM uses multiple orthogonal carriers for multi-carrier digital modulation
- The technique uses complex exponentials

$$x(t) = \sum_{m=0}^{N-1} X_m e^{j\frac{2\pi m}{T}t} \qquad 0 \le t \le T$$

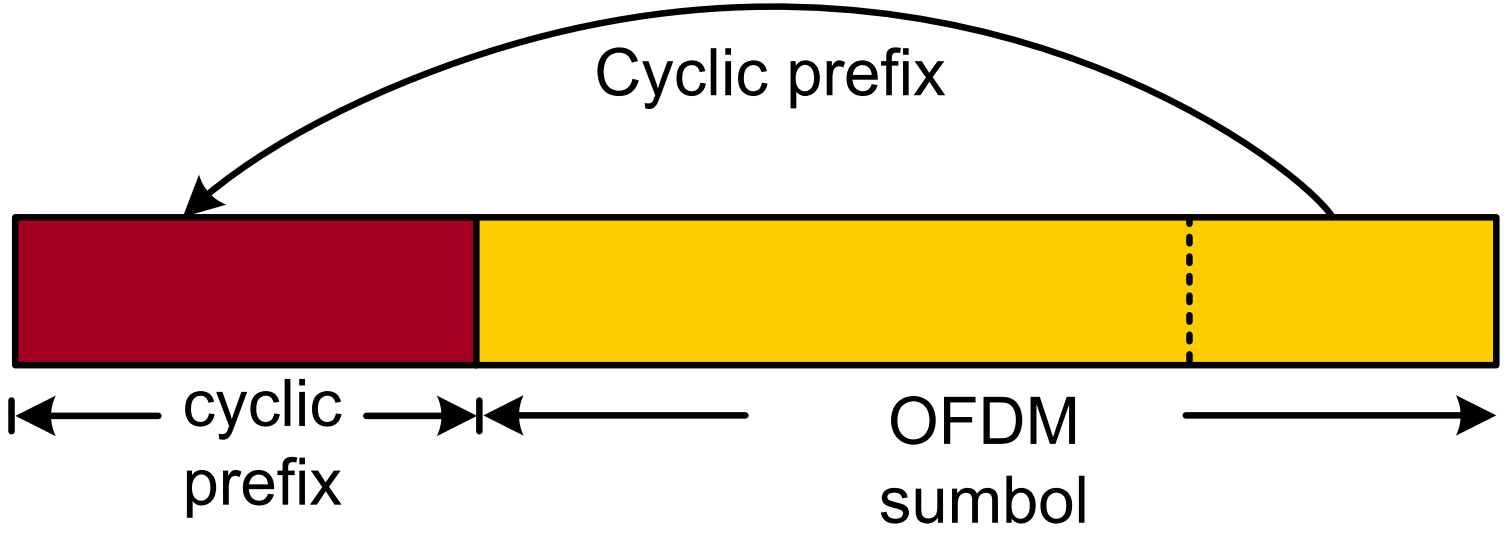
OFDM-based Digital Transmitter and Receiver

$$x(t) = \sum_{m=0}^{N-1} X_m e^{j\frac{2\pi m}{T}t} \qquad 0 \le t \le T$$

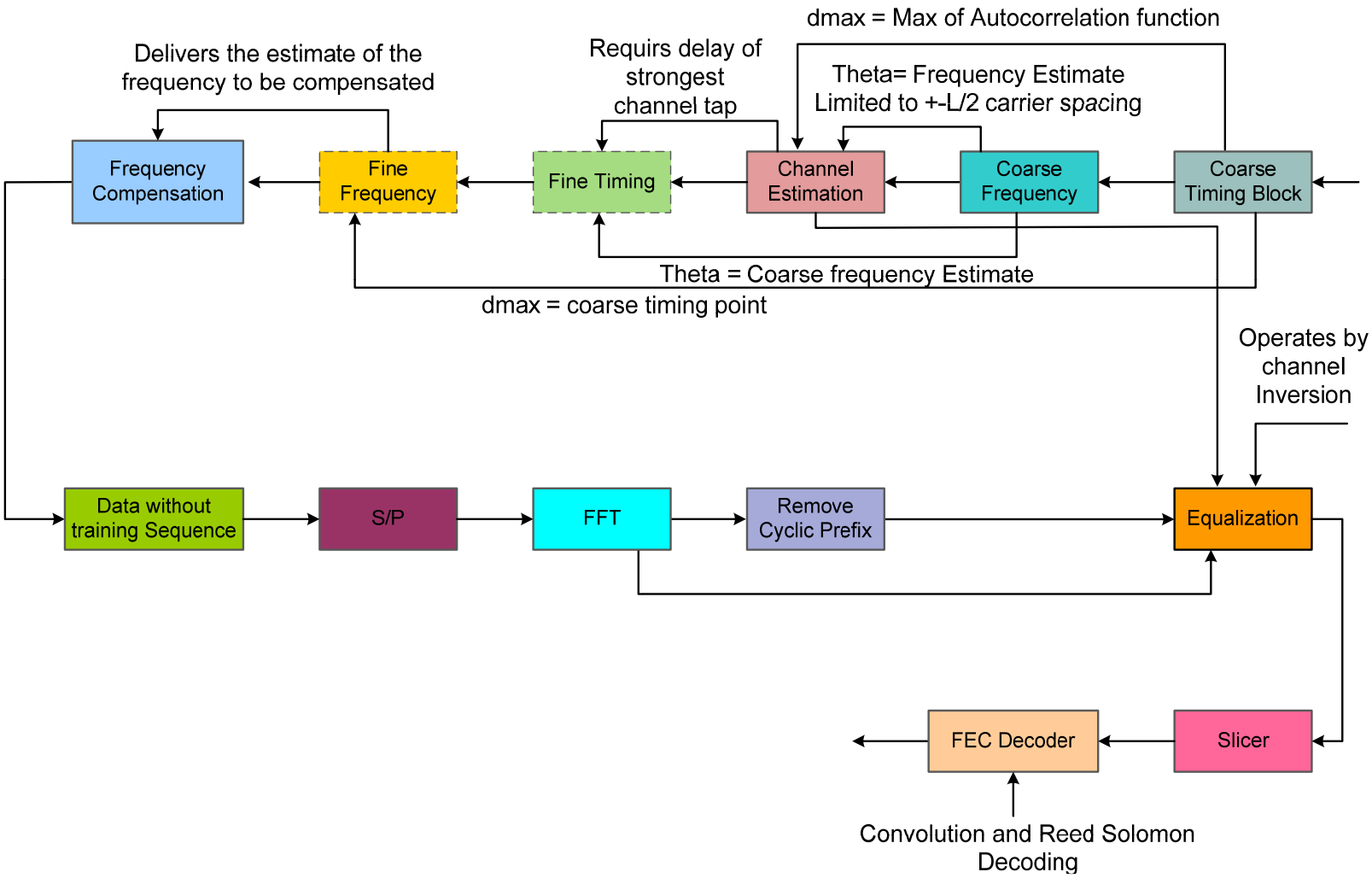
$$e^{j\frac{2\pi m}{T}t}$$

$$\frac{1}{T}\int_0^T e^{j\frac{2\pi l}{T}t} e^{-j\frac{2\pi m}{T}t} dt = \delta_{lm}$$

Adding cyclic prefix to an OFDM symbol



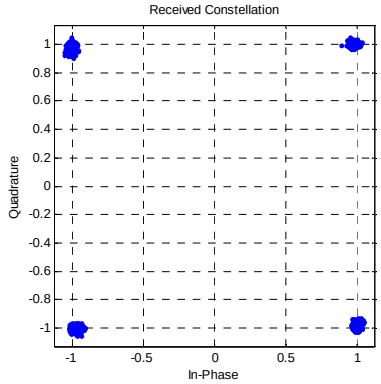
Block diagram of complete OFDM receiver



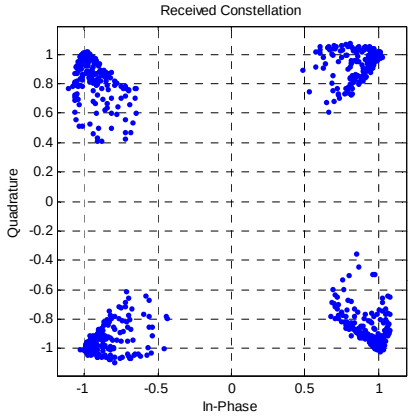
Digital Up-conversion and Mixing

- The modulated signal is to be digitally mixed with an intermediate frequency (IF) carrier
- In a digital receiver one of the two frequencies, 70 MHz or 21.4 MHz, is commonly used. The digital carrier is generated adhering to the Nyquist sampling criterion

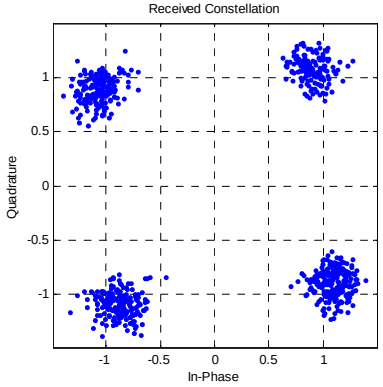
Constellation of QPSK-basedOFDMreceived signal for different channel conditions and frequency offsets. (a) 40 dB SNR. (b) 40 dB SNR with 30 Hz Doppler shift. (c) 20 dB SNR with 5 Hz
Doppler shift. (d) 10 dB SNR



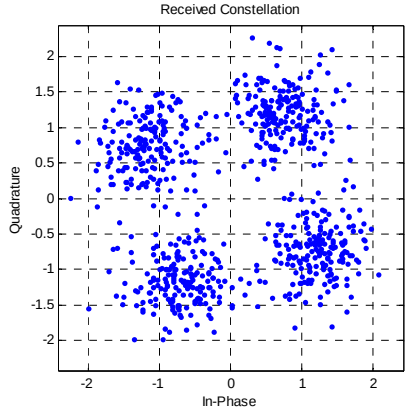
(a)



(b)

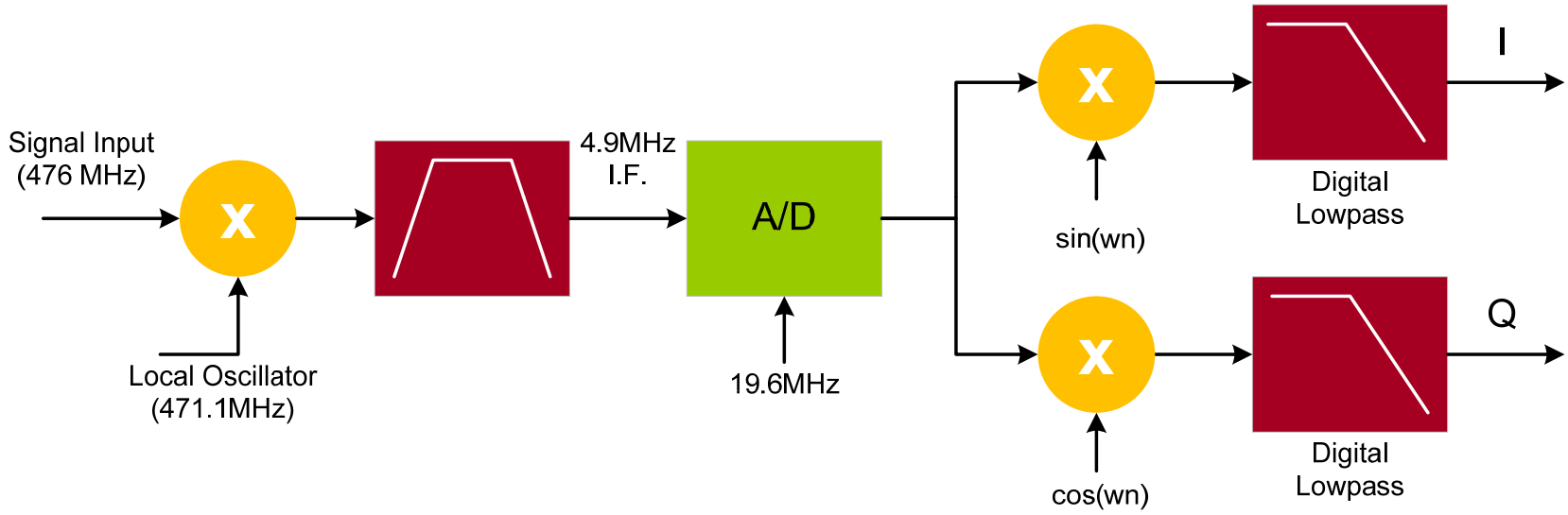


(c)



(d)

Front End of a Receiver



Questions/Feedback