



## + Chương 3

Tổng quan về máy tính và hệ thống kết nối  
trong máy tính



# Chương 3. Tổng quan về máy tính và hệ thống kết nối trong máy tính



## Phần I. Tổng quan về máy tính

### 3.1 Các thành phần của máy tính

### 3.2 Hoạt động của máy tính

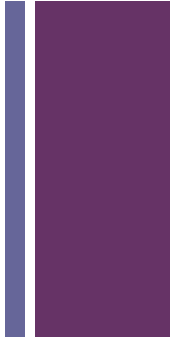
## Phần II. Hệ thống kết nối

### 3.3 Cấu trúc kết nối

### 3.4 Hệ thống bus

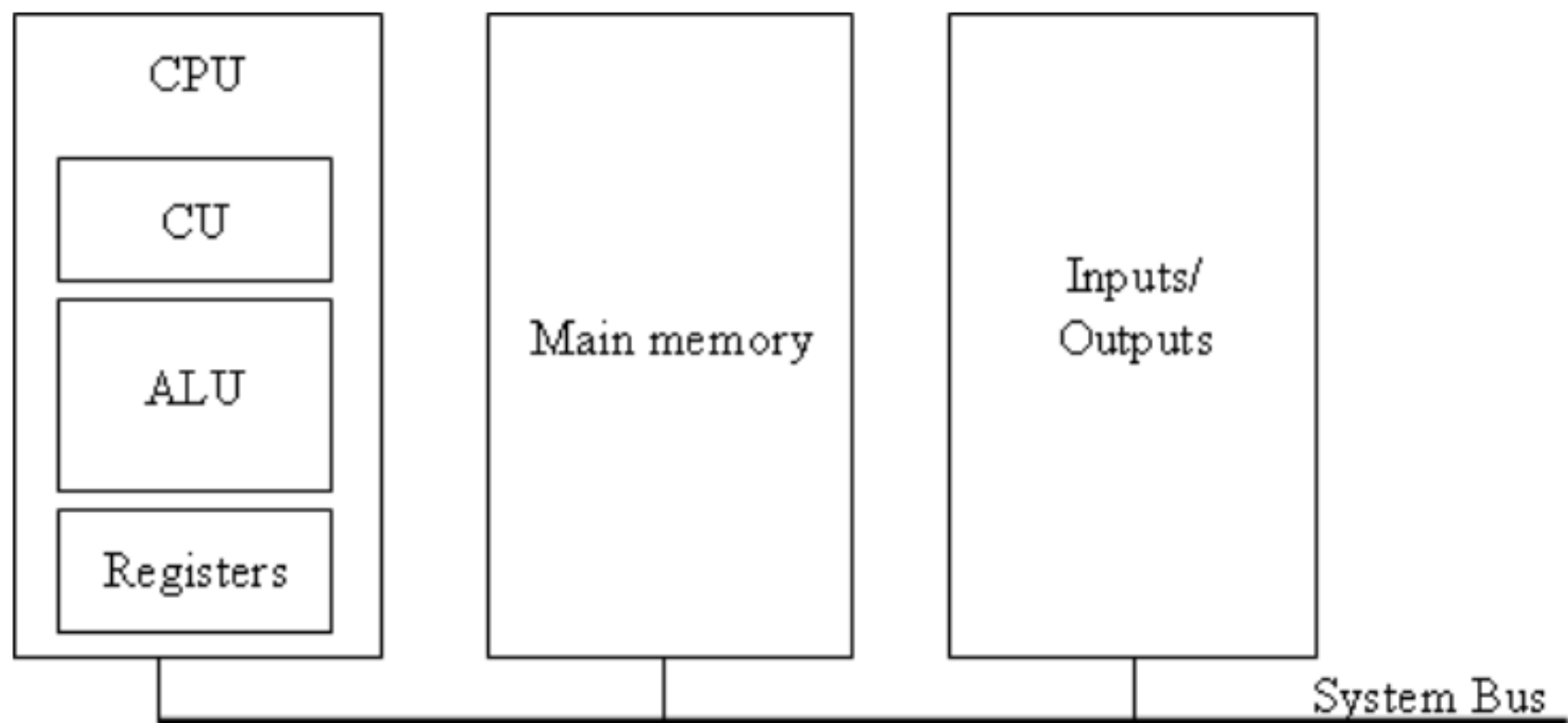
### 3.5 Kết nối điểm-điểm (Point-To-Point)

### 3.6 PCI Express



## 3.1. Các thành phần của máy tính

- Máy tính hiện đại ngày nay được thiết kế dựa trên **kiến trúc von Neumann** (Viện nghiên cứu Princeton)
- Kiến trúc Von Neumann có 3 điểm chính:
  - Dữ liệu và lệnh được lưu trữ trên cùng một bộ nhớ đọc-ghi (RAM)
  - Nội dung của dữ liệu được định vị theo vị trí (địa chỉ) mà không phụ thuộc vào kiểu dữ liệu.
  - Các lệnh được thực thi một cách tuần tự (trừ trong một số trường hợp yêu cầu gọi đến câu lệnh khác).



Kiến trúc von-Neumann hiện đại



# Các thành phần của máy tính (tiếp)



## ■ Phần mềm

- Một chuỗi các lệnh
- Khối CU làm chức năng phiên dịch từng lệnh và tạo ra tín hiệu điều khiển
- Quá trình thực hiện chương trình là truy xuất lệnh từ bộ nhớ và thực thi lệnh của CPU

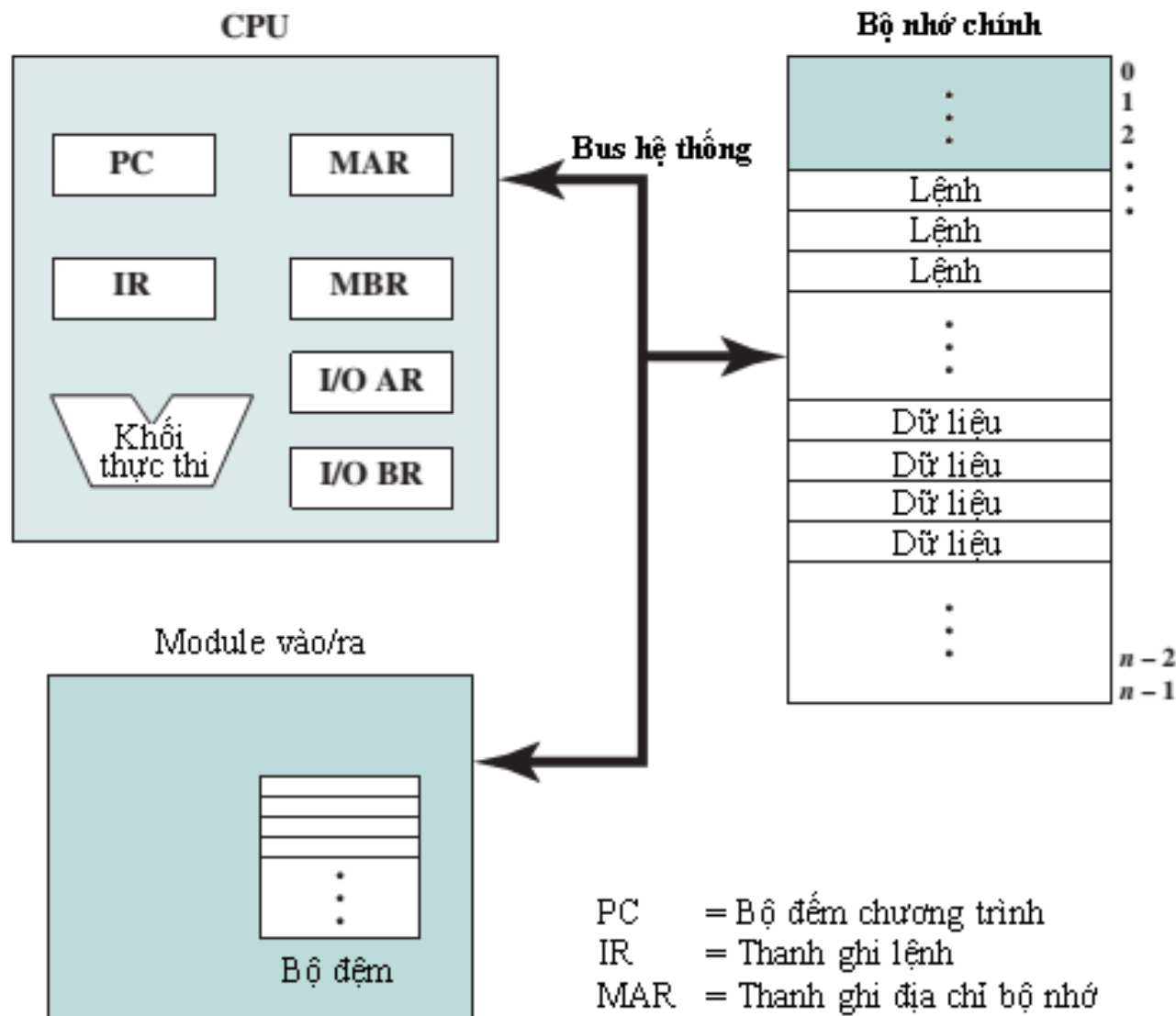
## ■ Phần cứng (3 thành phần chính)

### ■ CPU

- CU: Khối điều khiển thực hiện chức năng biên dịch và thực thi lệnh
- ALU: Khối tính toán số học và logic

### ■ Các Module vào/ra (I/O module)

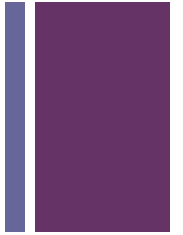
- Module vào: bao gồm các thành phần cơ bản cho việc nhận vào dữ liệu và lệnh; chuyển đổi chúng thành dạng tín hiệu sử dụng bên trong hệ thống
- Module ra: công cụ để hiển thị kết quả
- Bộ nhớ trong (bộ nhớ chính): bộ nhớ ROM, RAM: lưu trữ lệnh, dữ liệu
  - Bộ nhớ Cache: cải thiện hiệu suất của hệ thống



## Các thành phần của máy tính

PC = Bộ đếm chương trình  
 IR = Thanh ghi lệnh  
 MAR = Thanh ghi địa chỉ bộ nhớ  
 MBR = Thanh ghi đệm bộ nhớ  
 I/O AR = Thanh ghi địa chỉ vào/ra  
 I/O BR = Thanh ghi đệm vào/ra

## + Giải thích một số thanh ghi trong hình:



- Thanh ghi **MAR (Memory Address Register)** chứa địa chỉ trong bộ nhớ cho lần đọc hoặc ghi tiếp theo
- Thanh ghi **MBR (Memory Buffer Register)** dữ liệu được ghi vào bộ nhớ hoặc nhận dữ liệu được đọc từ bộ nhớ.
- Thanh ghi **I/OAR (I/O Address Register)** xác định một thiết bị I/O cụ thể.
- Thanh ghi **I/O BR (I/O Buffer Register)** được sử dụng để trao đổi dữ liệu giữa một mô-đun I/O và CPU.
- Thanh ghi **PC (Program counter Register)** chứa địa chỉ lệnh tiếp theo
- Thanh ghi **IR (Instruction Register)** chứa lệnh đang được thực thi

## + 3.2. Hoạt động của máy tính

- Hoạt động cơ bản của máy tính là thực hiện chương trình:
  - a. Thực hiện lệnh: chu kỳ lệnh
  - b. Thực hiện lệnh có xử lý ngắt
  - c. Thực hiện các chức năng vào ra





## 3.2. Hoạt động của máy tính



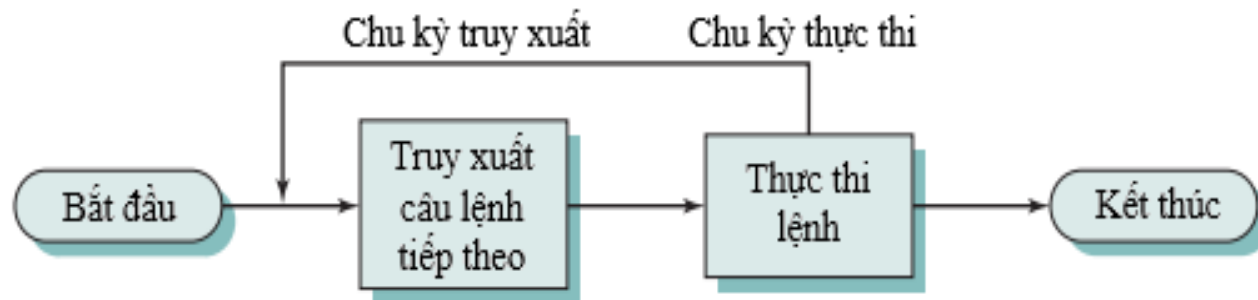
a. Thực hiện lệnh: chu kỳ lệnh

- Chức năng chính của máy tính là thực thi chương trình (một tập lệnh lưu trữ trong BN): VXL phải thực hiện lần lượt các lệnh
- Quá trình VXL thực hiện 1 lệnh gồm 2 bước: lấy lệnh (truy xuất) từ bộ nhớ và thực thi lệnh.
- Việc thực thi một chương trình là quá trình lặp đi lặp lại việc truy xuất và thực thi lệnh
- Quá trình thực hiện một lệnh được gọi là *chu kỳ lệnh (instruction cycle)*
  - Quá trình truy xuất lệnh từ bộ nhớ được gọi là *chu kỳ truy xuất (fetch cycle)*
  - Quá trình thực thi lệnh được gọi là *chu kỳ thực thi (execute cycle)*

# + a. Truy xuất và thực thi lệnh

## Chu kỳ truy xuất

- Vào đầu mỗi chu kỳ lệnh, bộ xử lý truy xuất một lệnh từ bộ nhớ
- Thanh ghi PC (Program Counter) giữ địa chỉ của lệnh được truy xuất tiếp theo
- Bộ xử lý tăng PC sau mỗi lần truy xuất lệnh do đó nó sẽ truy xuất được lệnh tiếp theo vào lần sau.
- Lệnh vừa được truy xuất được tải vào thanh ghi IR (Instruction Register)
- Bộ xử lý biên dịch lệnh và thi hành những hành động cần thiết



Chu kỳ lệnh cơ bản



# a. Truy xuất và thực thi lệnh

## Chu kỳ thực thi

- CPU giải mã và thực hiện các hoạt động (action) tương ứng được chỉ ra trong mã lệnh (Opcode)
- Có 4 nhóm hoạt động chính của một CPU:

### Bộ xử lý – bộ nhớ

- Dữ liệu truyền từ bộ xử lý đến bộ nhớ hoặc ngược lại

### Bộ xử lý – I/O

- Dữ liệu truyền đến/đi từ thiết bị ngoại vi bằng cách truyền thông tin giữa bộ xử lý và module I/O

### Xử lý dữ liệu

- Bộ xử lý có thể thực hiện một số phép toán số học hoặc logic trên dữ liệu

### Điều khiển

- Đưa ra lệnh chỉ rõ thứ tự thực hiện các lệnh bị thay đổi

# + Ví dụ việc thực hiện lệnh

Máy giả thiết gồm một số thông tin cấu hình như sau:



(a) Định dạng lệnh



(b) Định dạng số nguyên

Thanh ghi PC = Địa chỉ của lệnh

Thanh ghi IR = Lệnh sẽ được thực hiện

Thanh ghi AC = Thanh ghi lưu trữ dữ liệu tạm thời

(c) Các thanh ghi trong CPU

0001 = Đọc dữ liệu từ bộ nhớ vào thanh ghi AC

0010 = Lưu trữ dữ liệu từ thanh ghi AC vào bộ nhớ

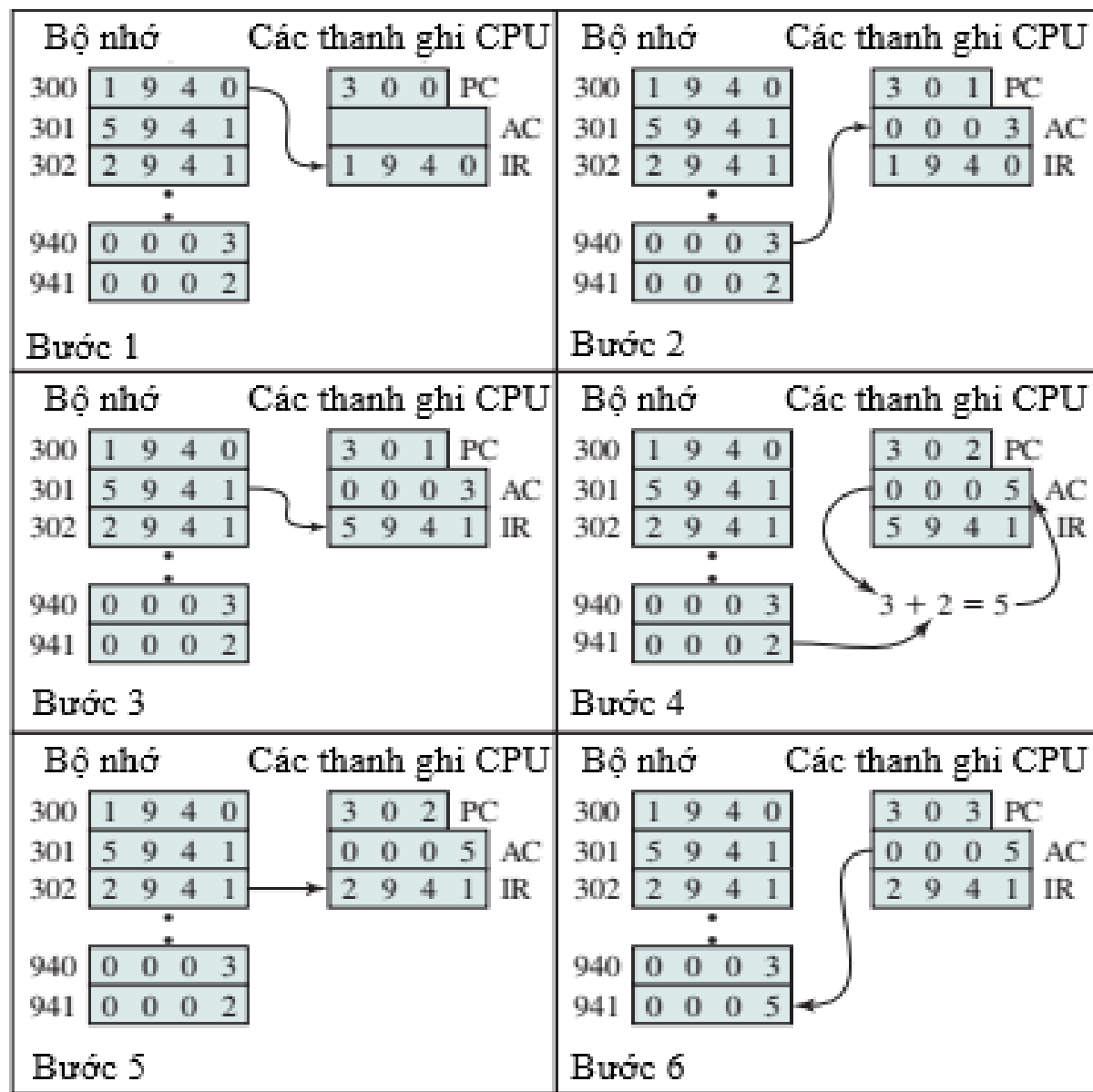
0101 = Đọc dữ liệu từ bộ nhớ và cộng vào thanh ghi AC

(d) Các mã lệnh được sử dụng

Hình 3.4 Đặc tính của máy giả thiết



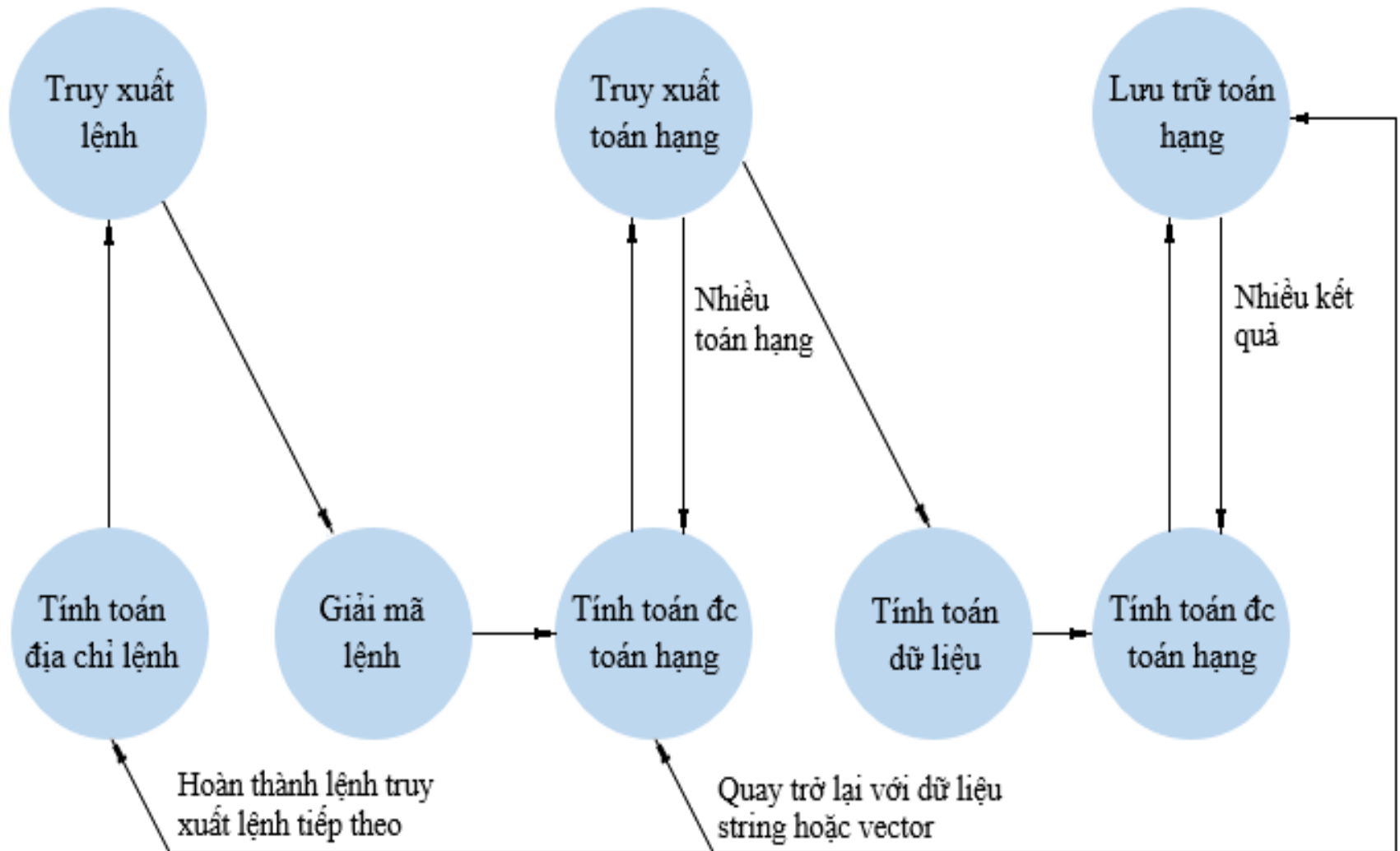
## Ví dụ Thực hiện lệnh



Dữ liệu và lệnh được biểu diễn dưới dạng mã thập lục phân

+

# Sơ đồ trạng thái chu kỳ lệnh



## + Ví dụ

- Máy giả thiết trong ví dụ trên có hai lệnh vào/ra sau:

0011 = Đọc dữ liệu từ module I/O vào thanh ghi AC

0111 = Ghi dữ liệu từ AC ra module I/O

Biết các thiết bị ngoại vi được đánh địa chỉ 12b. Giải thích hoạt động của chương trình sau (giống Ví dụ 3.5):

1. Đọc dữ liệu từ thiết bị 5 vào thanh ghi AC.
2. Cộng AC với địa chỉ 940 của bộ nhớ.
3. Ghi AC ra thiết bị 6.

Giả sử giá trị được lấy từ thiết bị 5 là 5 và địa chỉ bộ nhớ 940 có giá trị 7.

## + 3.2. Hoạt động của máy tính

### b. Xử lý ngắt

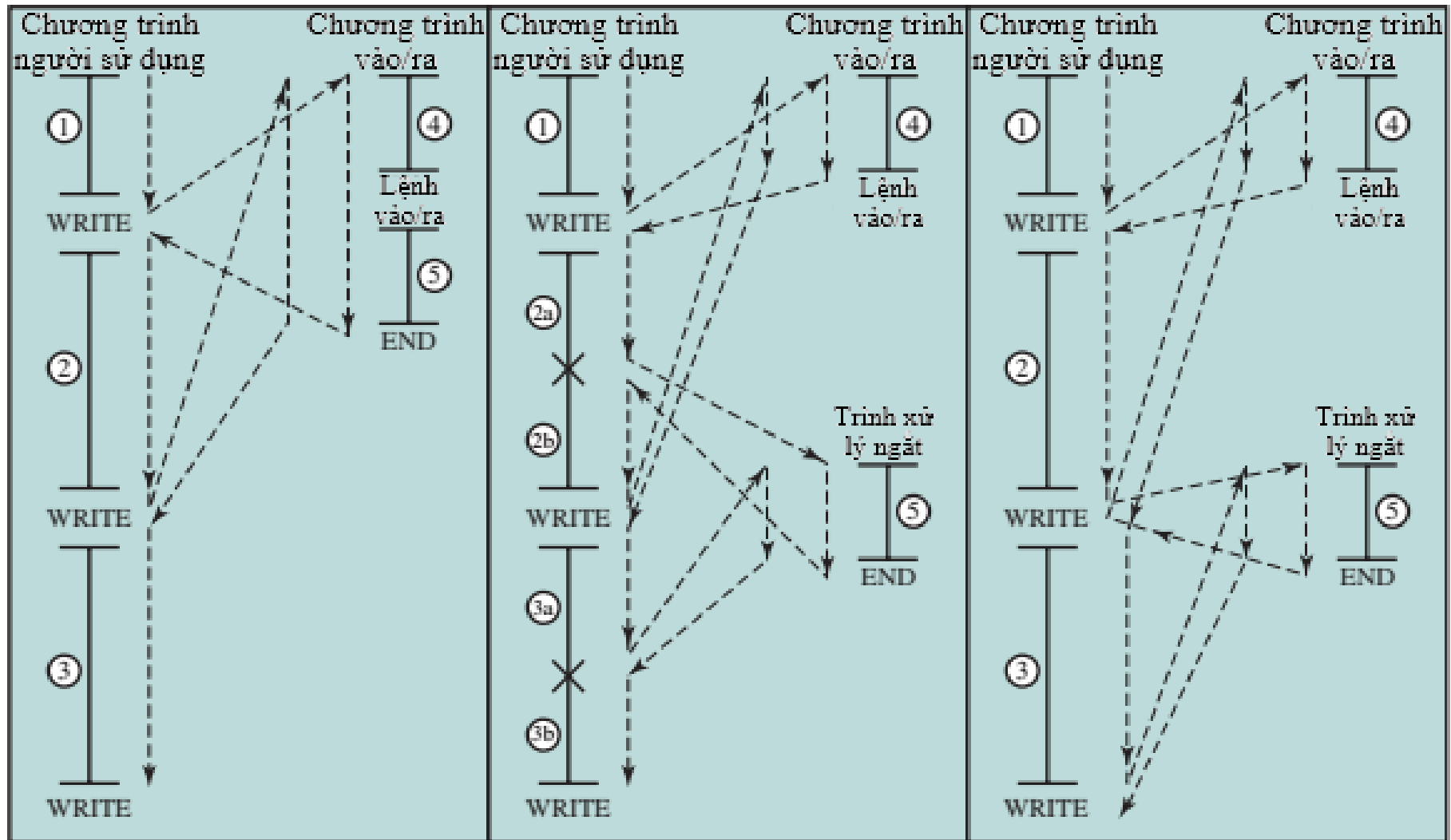
- **Ngắt** là một cơ chế máy tính cho phép các module khác (I/O, bộ nhớ có thể ngắt quá trình xử lý thông thường của BXL. Một số ngắt:
  - **Ngắt chương trình:** Sinh ra bởi lỗi thi hành lệnh, ví dụ như tràn số học, lỗi chia cho 0, cố tình thực hiện các lệnh máy không hợp lệ, hoặc tham chiếu ngoài phạm vi bộ nhớ mà người sử dụng được phép
  - **Ngắt định thời:** Sinh ra bởi đồng hồ nằm trong bộ xử lý. Nó cho phép hệ điều hành thực hiện các chức năng cơ bản nhất định.
  - **Ngắt I/O:** Sinh ra bởi bộ điều khiển I/O, để báo hiệu hoàn thành một thao tác, yêu cầu dịch vụ từ bộ xử lý, hoặc báo hiệu các trường hợp lỗi
  - **Gián đoạn lỗi phần cứng:** Gây ra bởi một số lỗi như lỗi nguồn hay lỗi bộ nhớ



# + Quá trình xử lý ngắt

- Ngắt được đưa vào chủ yếu như là 1 cách để để cải thiện hiệu quả xử lý:
  - Trong trường hợp VXL thực hiện chương trình có trao đổi dữ liệu với I/O
  - Do tốc độ của I/O chậm hơn rất nhiều so với VXL → VXL phải đợi I/O
  - Giải pháp: trong lúc chờ đợi I/O, VXL thực hiện tiếp các phần công việc khác → đến khi I/O xong, nó sẽ gửi tín hiệu đến VXL (tín hiệu y/c ngắt) → VXL dừng công việc đang làm (ngắt), phục vụ I/O → VXL tiếp tục cv đang thực hiện

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

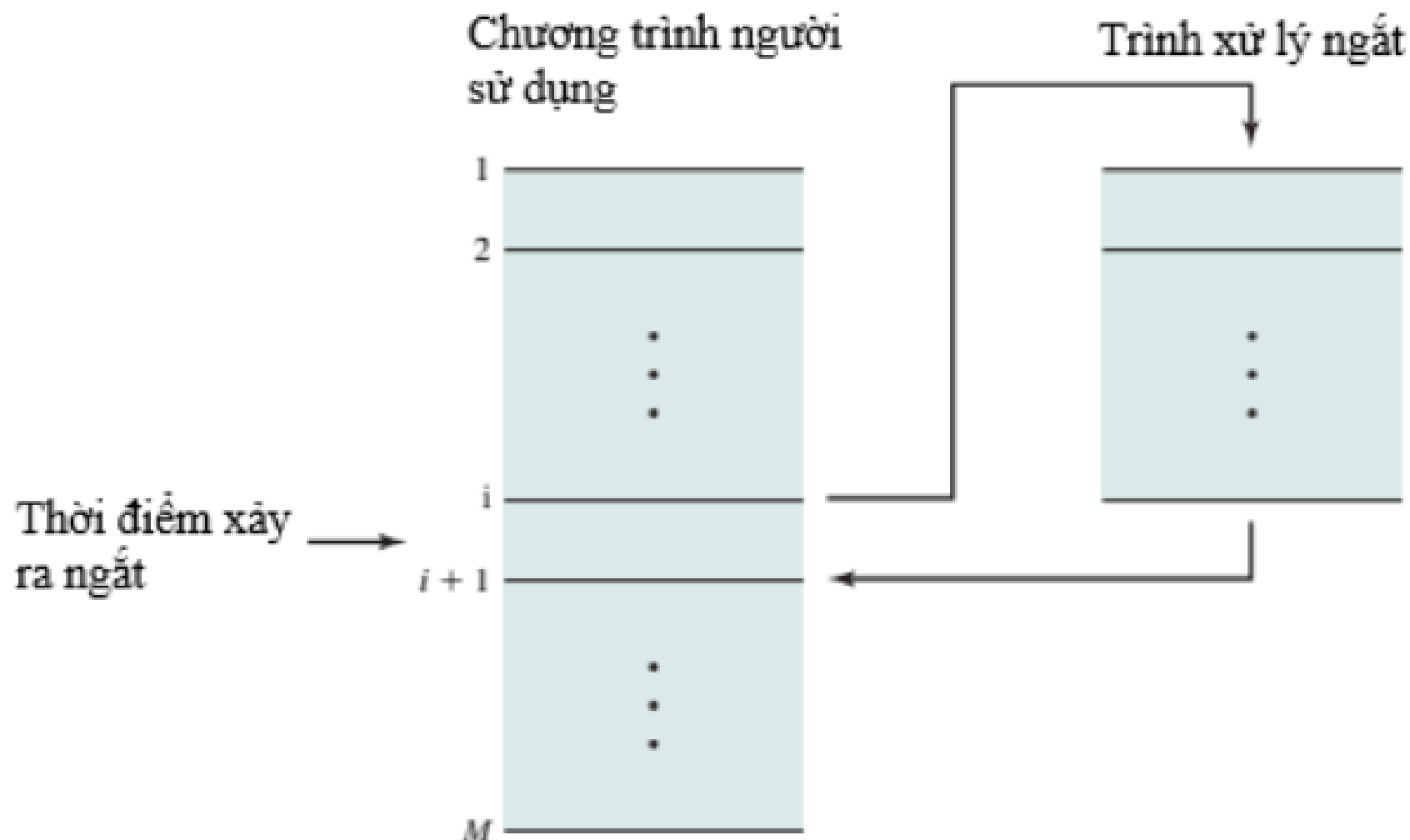


a) Không có ngắt

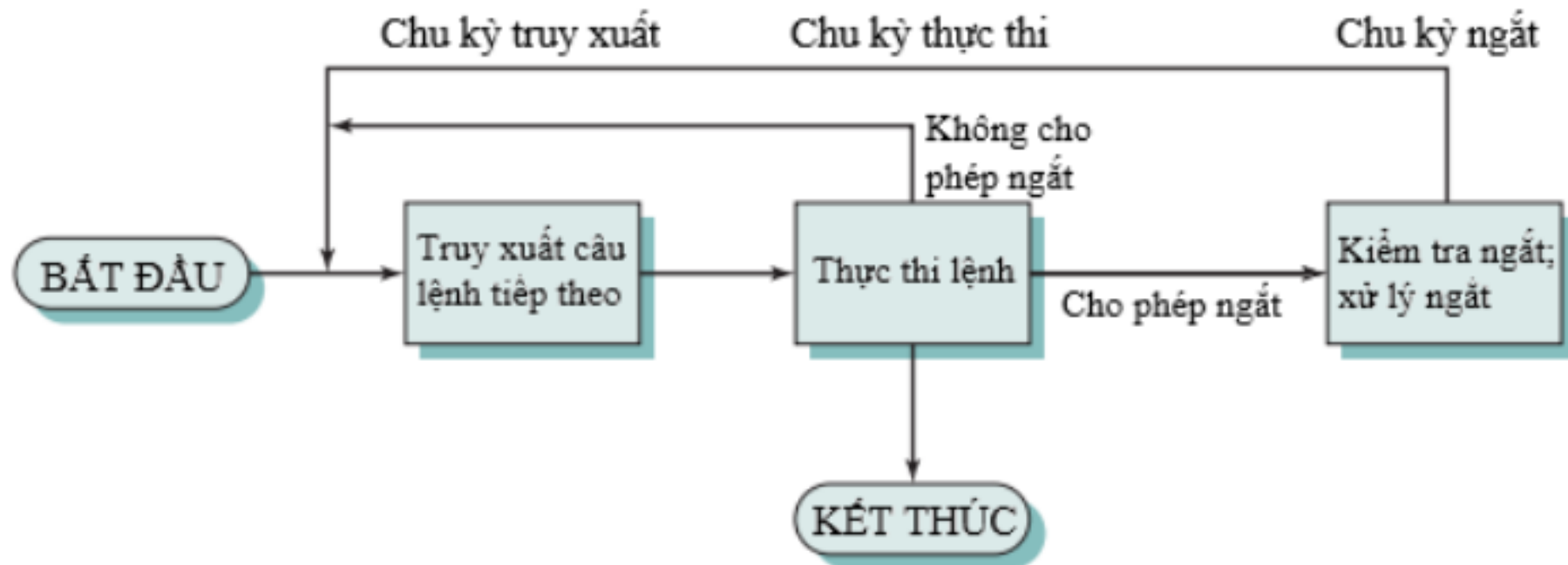
b) Sử dụng ngắt, đọi vào/ra ngắn

c) Sử dụng ngắt, đợi vào/ra dài

# + Điều khiển ngắt



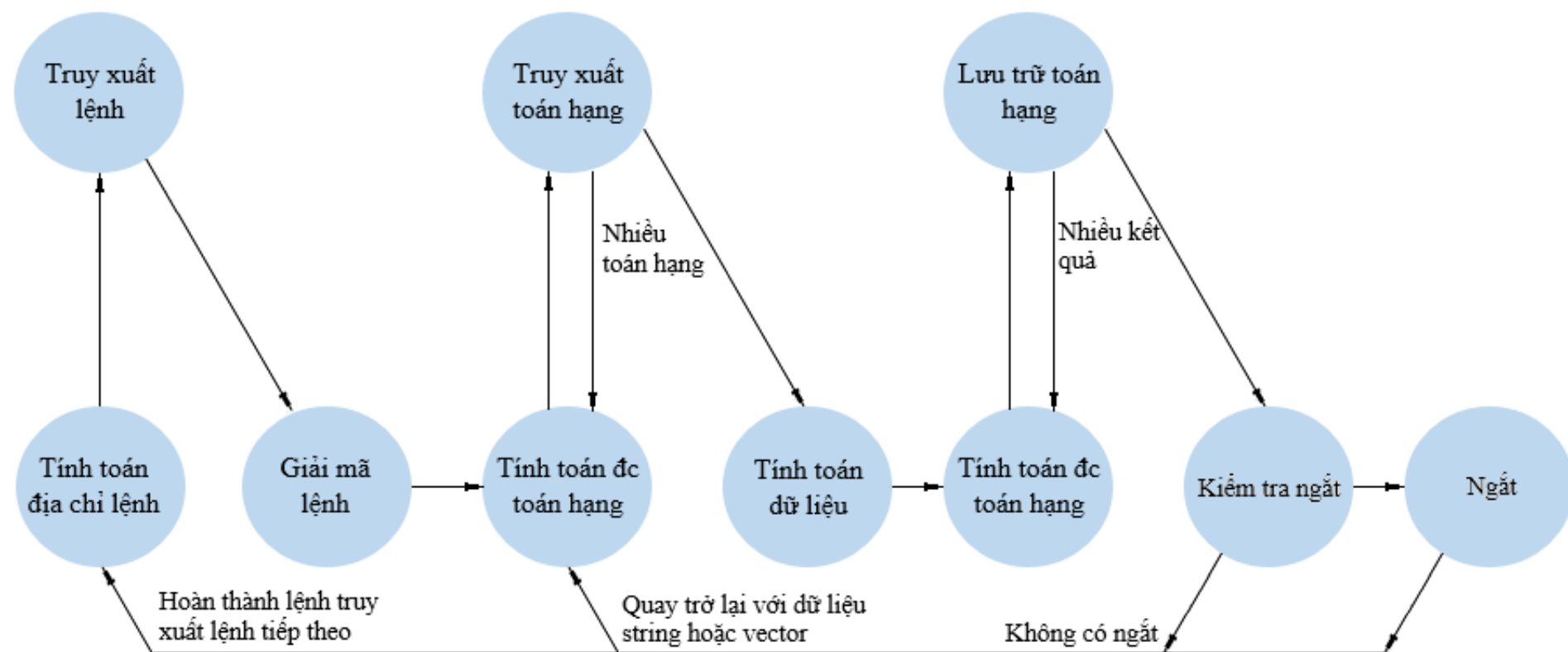
# + Chu kỳ lệnh có ngắt



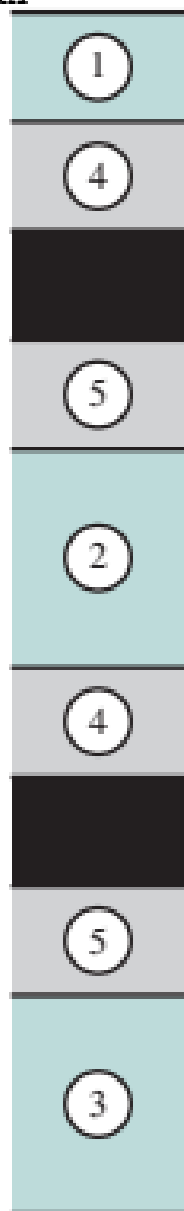
Hình 3.9 Chu kỳ lệnh có ngắt

# Sơ đồ trạng thái chu kỳ lệnh

## Có ngắt



Thời gian



Hoạt động vào/ra;  
bộ xử lý đợi

Hoạt động vào/ra;  
bộ xử lý đợi

(a) Không có ngắt



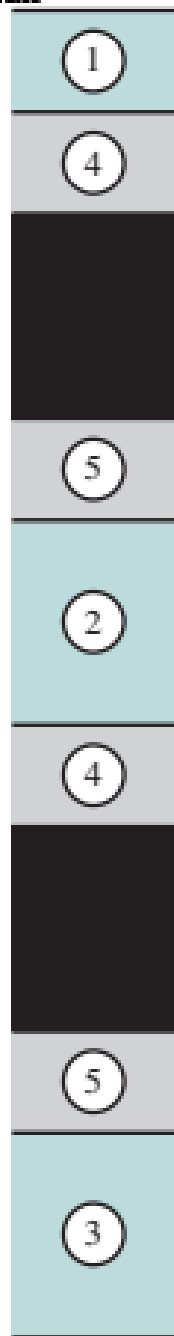
Hoạt động vào ra  
cùng lúc với bxl  
thực thi

Hoạt động vào ra  
cùng lúc với bxl  
thực thi

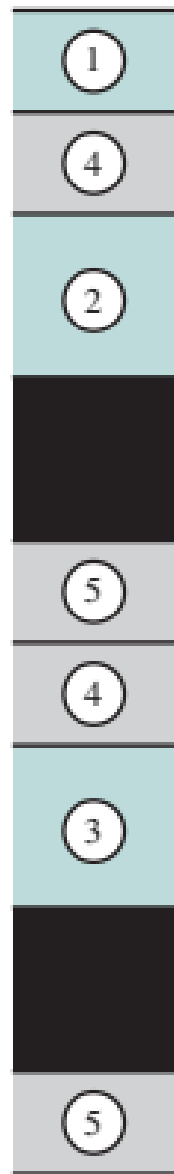
(b) Có ngắt

Minh họa  
thời gian  
thực hiện  
chương  
trình

Thời gian



(a) Không có ngắt



(b) Có ngắt

Định thời  
chương  
trình:  
Đợi I/O dài



# Xử lý nhiều ngắt

Trong trường hợp có nhiều ngắt, hai phương pháp để xử lý:

## ■ Tắt ngắt

- Bộ vi xử lý sẽ bỏ qua các tín hiệu ngắt khác trong khi xử lý một yêu cầu ngắt
- Các yêu cầu ngắt đó sẽ phải chờ đến khi bộ xử lý xử lý xong ngắt hiện tại
- Sau khi thực thi xong một ngắt, bộ xử lý sẽ kiểm tra xem có ngắt nào đang chờ không. Các ngắt sẽ được xử lý lần lượt

## ■ Xác định ưu tiên

- Cho phép các ngắt có mức độ ưu tiên cao hơn được ngắt các ngắt có mức độ ưu tiên thấp hơn



Chương trình  
người sử dụng



Trình xử lý  
ngắt X



Trình xử lý  
ngắt Y



(a) Xử lý ngắt tuần tự

Chương trình  
người sử dụng



Trình xử lý  
ngắt X



Trình xử lý  
ngắt Y

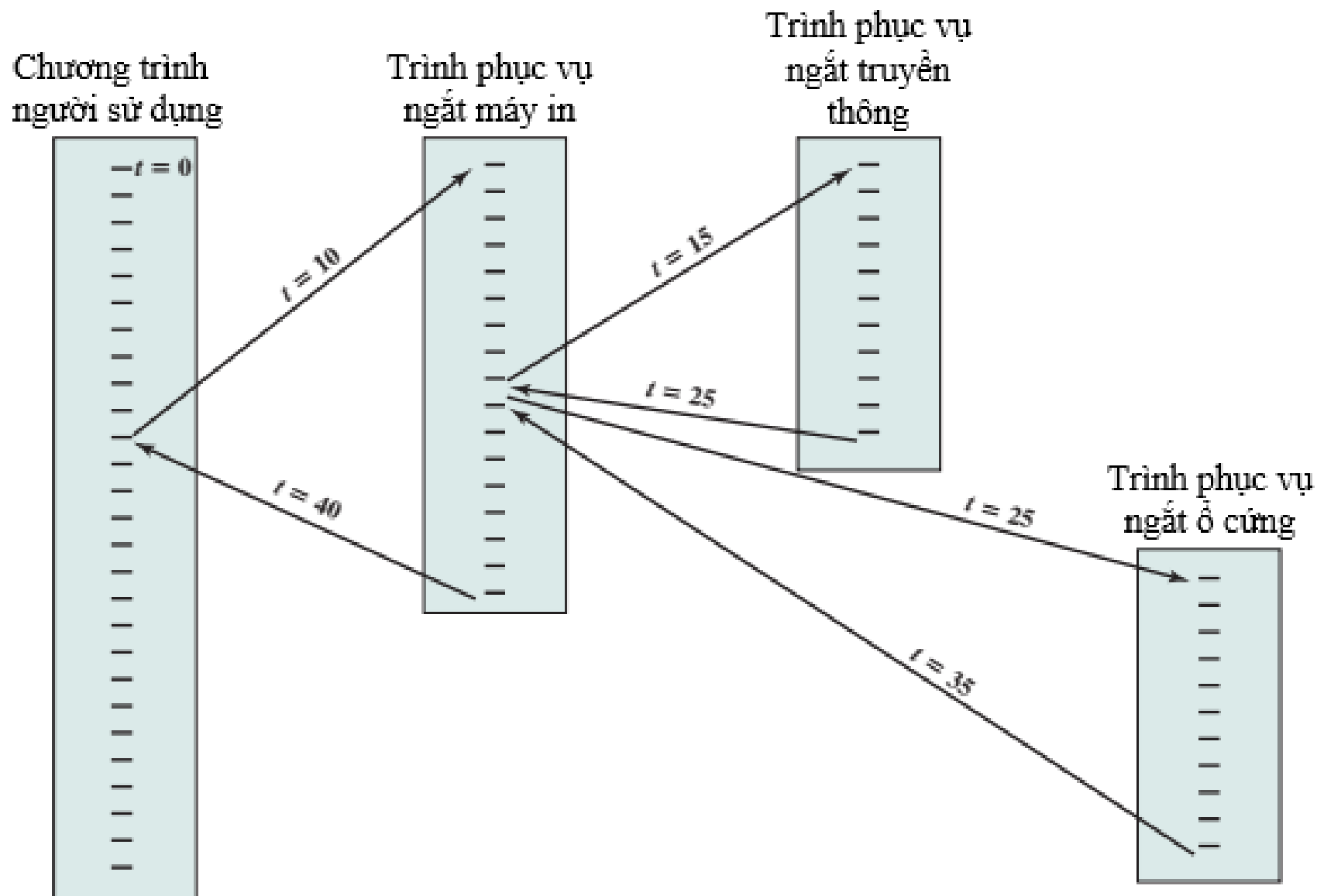


(b) Xử lý ngắt ưu tiên

Điều khiển  
ngắt

Nhiều ngắt

# + Trình tự thời gian của xử lý nhiều ngắt



## + c. Chức năng I/O

- Module I/O có thể chuyển dữ liệu trực tiếp với bộ xử lý
- Bộ xử lý có thể đọc dữ liệu từ hoặc ghi dữ liệu lên module I/O
  - Bộ xử lý xác định thiết bị nào được điều khiển bởi module I/O nào
  - Khi làm việc với module I/O, một chuỗi lệnh tương tự như P13 có thể được thực hiện chỉ khác ở các lệnh I/O chứ không phải là các lệnh tham chiếu đến bộ nhớ
- Cơ chế truy cập bộ nhớ trực tiếp (Direct Memory Access - DMA): cho phép I/O trao đổi dữ liệu trực tiếp với bộ nhớ
  - Bộ xử lý cấp cho module I/O quyền đọc/ghi vào bộ nhớ do đó việc truyền tin giữa module I/O và bộ nhớ có thể diễn ra trực tiếp mà không cần thông qua bộ xử lý
  - Giải phóng bộ XL khỏi nhiệm vụ điều khiển việc chuyển dữ liệu

## + 3.3 Cấu trúc kết nối

- Các thành phần chính (bộ vi xử lý, bộ nhớ, I /O) của máy tính cần được kết nối để trao đổi dữ liệu với nhau
- Một tập các đường kết nối tạo thành *cấu trúc kết nối* (Interconnection Structures)
- Cấu trúc này được thiết kế phụ thuộc vào cơ chế trao đổi dữ liệu giữa các thành phần máy tính.

# Cấu trúc kết nối hỗ trợ các hình thức truyền sau:



Bộ nhớ  
tới bộ  
xử lý

Bộ xử lý  
đọc 1 lệnh  
hoặc 1  
đơn vị dữ  
liệu từ bộ  
nhớ

Bộ xử lý  
tới bộ  
nhớ

Bộ xử lý  
ghi 1 đơn  
vị dữ liệu  
vào bộ  
nhớ

I/O tới  
bộ xử lý

Bộ xử lý  
đọc dữ  
liệu từ  
thiết bị  
I/O thông  
qua I/O  
module

Bộ xử lý  
tới I/O

Bộ xử lý  
gửi dữ  
liệu tới  
thiết bị  
I/O

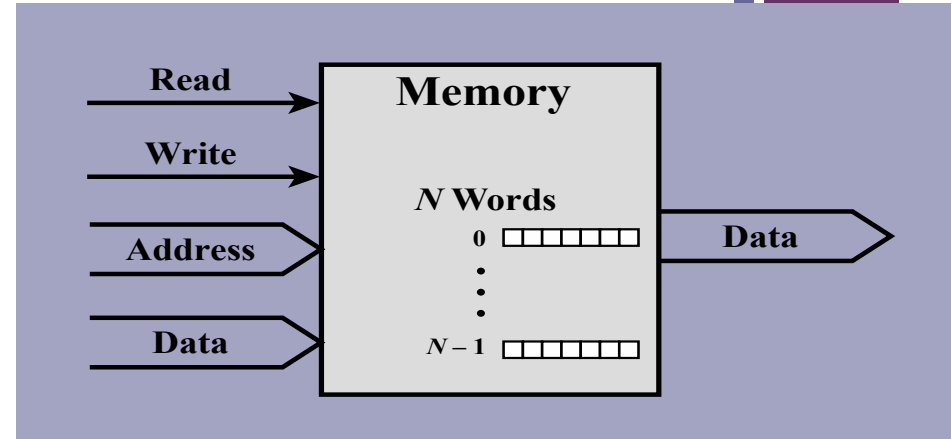
I/O  
tới/từ  
bộ nhớ

Module  
I/O được  
phép trao  
đổi dữ liệu  
trực tiếp  
với bộ nhớ  
mà không  
cần đi qua  
bộ xử lý  
nhờ DMA

# + Các dạng dữ liệu đến/đi từ các thành phần máy tính

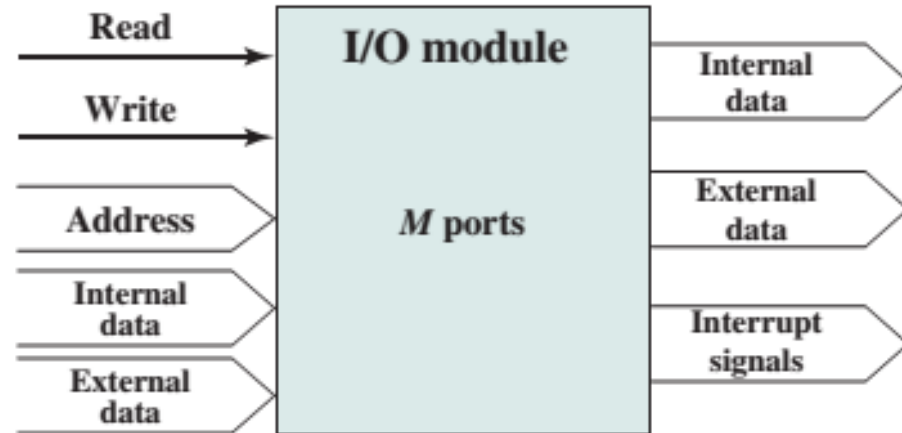
## ■ Bộ nhớ:

- T/h điều khiển đọc/ghi
- T/h địa chỉ
- Dữ liệu đi vào/ra



## ■ Module I/O

- T/h điều khiển đọc/ghi
- T/h địa chỉ
- Dữ liệu bên trong (ngoài): dữ liệu đến từ CPU(thiết bị ngoại vi)
- Tín hiệu ngắt: phát ra từ module I/O gửi đến CPU

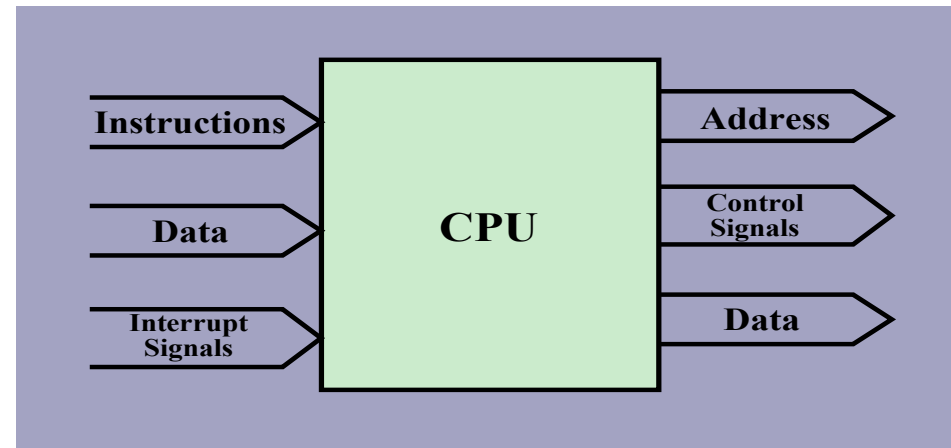




# Các dạng dữ liệu đến/đi từ các thành phần máy tính

## ■ CPU

- Lệnh: truy xuất từ bộ nhớ
- Dữ liệu đến hoặc đi từ bộ nhớ hoặc I/O
- T/h ngắt: do I/O module gửi tới
- T/h địa chỉ: định vị một ô nhớ hoặc một tb ngoại vi
- T/h điều khiển



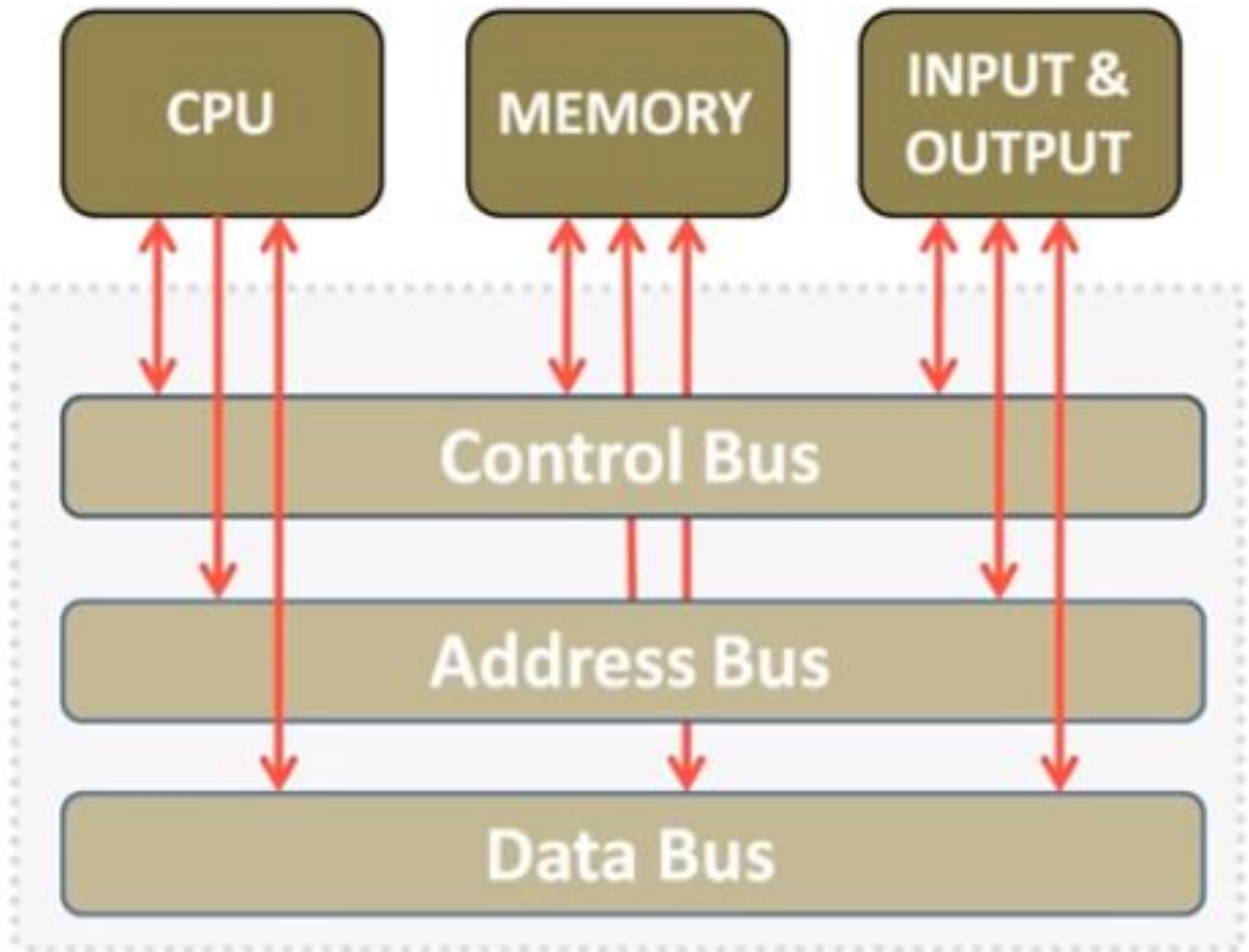
## + Một số cấu trúc kết nối

- Cấu trúc kết nối phổ biến nhất: cấu trúc bus và cấu trúc đa bus (phần 3.4)
- Cấu trúc kết nối điểm – điểm: QPI (phần 3.5) và PCIe (phần 3.6)



## + 3.4 Kết nối Bus (hệ thống Bus)

- **Bus:** đường thông tin kết nối giữa 2 hay nhiều thiết bị.
- Là đường truyền chia sẻ: Tín hiệu truyền bởi 1 thiết bị bất kì có thể được nhận bởi tất cả các thiết bị khác kết nối với bus đó
  - Nếu 2 thiết bị cùng truyền 1 lúc, tín hiệu của chúng sẽ bị chồng nhau và bị méo
- Một bus thường gồm nhiều đường, mỗi đường có khả năng truyền tín hiệu dưới dạng bit 1 và bit 0
- Hệ thống máy tính có một số loại bus khác nhau cung cấp đường kết nối giữa các thành phần thuộc các cấp khác nhau của hệ thống máy tính
- Bus hệ thống
  - Kết nối các thành phần chính của máy tính (bộ xử lý, bộ nhớ, module I/O)
  - Gồm 50 đến 100 đường:
    - *Data bus (bus dữ liệu): gồm các đường truyền dữ liệu*
    - *Address bus (bus địa chỉ): gồm các đường địa chỉ*
    - *Control bus (bus điều khiển): các đường truyền tín hiệu điều khiển*

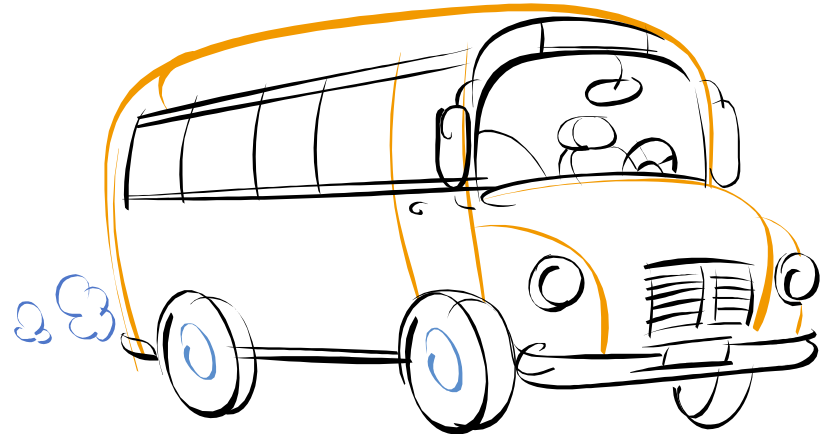




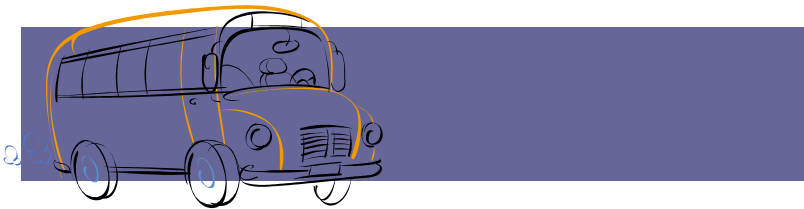
# a. Cấu trúc bus

## Bus dữ liệu (data bus)

- Gồm các đường dữ liệu (data lines) để truyền dữ liệu giữa các module hệ thống: **data bus**
- Data bus bao gồm 32, 64, 128 đường hay nhiều hơn
- Số lượng đường được xem là *độ rộng* của bus dữ liệu
- Số lượng đường nối quyết định bao nhiêu bit có thể truyền đi cùng một lúc
- Độ rộng bus dữ liệu là yếu tố chính quyết định hiệu suất toàn hệ thống

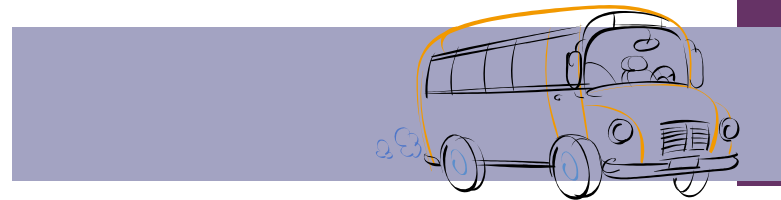


## + Bus địa chỉ



- Được sử dụng để xác định địa chỉ nguồn/đích của dữ liệu trên bus dữ liệu.
  - Nếu bộ xử lý muốn đọc 1 word từ bộ nhớ, nó sẽ đặt địa chỉ của word đó lên đường bus địa chỉ.
- Độ rộng bus xác định dung lượng nhớ tối đa của hệ thống
- Cũng được sử dụng để xác định cổng vào/ra (I/O port) trên module I/O.
  - Các bit cao được sử dụng để lựa chọn module cụ thể trên bus còn bit thấp dùng để chọn vị trí bộ nhớ hoặc cổng vào/ra trong module.

## Bus điều khiển



- Được sử dụng để điều khiển việc truy nhập và sử dụng dữ liệu và bus địa chỉ.
- Bởi vì dữ liệu và bus địa chỉ được chia sẻ cho tất cả các thành phần nên cần phải có một công cụ kiểm soát việc sử dụng chúng.
- Các tín hiệu điều khiển truyền cả thông tin lệnh và định thời giữa các mô đun hệ thống.
- Tín hiệu định thời xác định tính hợp lệ của dữ liệu và thông tin địa chỉ.
- Tín hiệu lệnh chỉ ra thao tác (operation) cần được thực hiện.
  - *VD: t/h điều khiển read/write*



# Ví dụ: đọc dữ liệu từ Bộ nhớ vào VXL

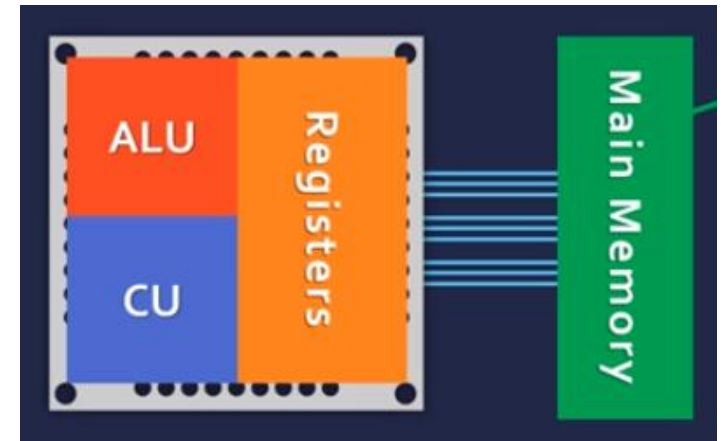
## ■ CPU gửi:

- Địa chỉ qua bus địa chỉ
- Tín hiệu yêu cầu đọc (READ) qua bus điều khiển

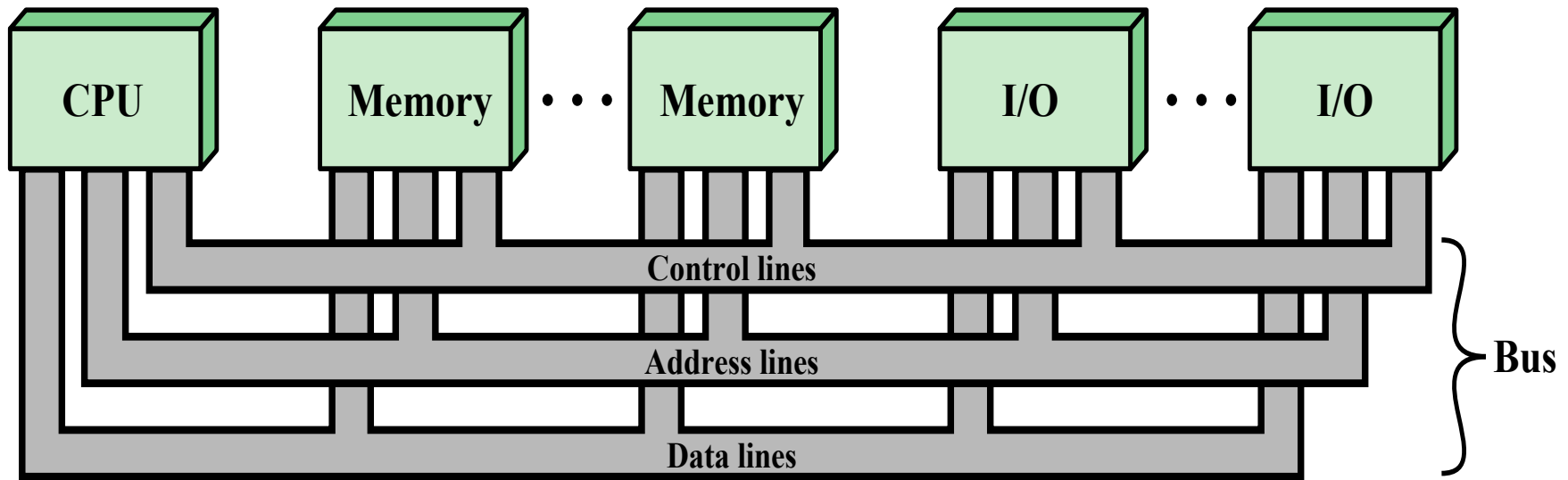
## ■ RAM:

- Nhận địa chỉ từ bus địa chỉ, giải mã địa chỉ
- Xác định yêu cầu: đọc dữ liệu
- Lấy dữ liệu từ ngăn nhớ đó đặt lên bus dữ liệu

- CPU: đọc dữ liệu từ bus dữ liệu, ghi vào thanh ghi. Loại bỏ các tín hiệu điều khiển và địa chỉ.



# Sơ đồ kết nối BUS

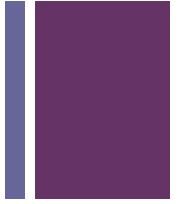


Hoạt động của bus như sau:

- Nếu một module muốn gửi dữ liệu đến một module khác nó phải làm hai việc: (1) yêu cầu việc sử dụng bus và (2) truyền dữ liệu qua bus.
- Nếu một module yêu cầu dữ liệu từ một module khác, nó phải (1) yêu cầu sử dụng bus và (2) chuyển yêu cầu tới module khác qua bus điều khiển và địa chỉ. Sau đó phải chờ cho module thứ hai gửi dữ liệu.

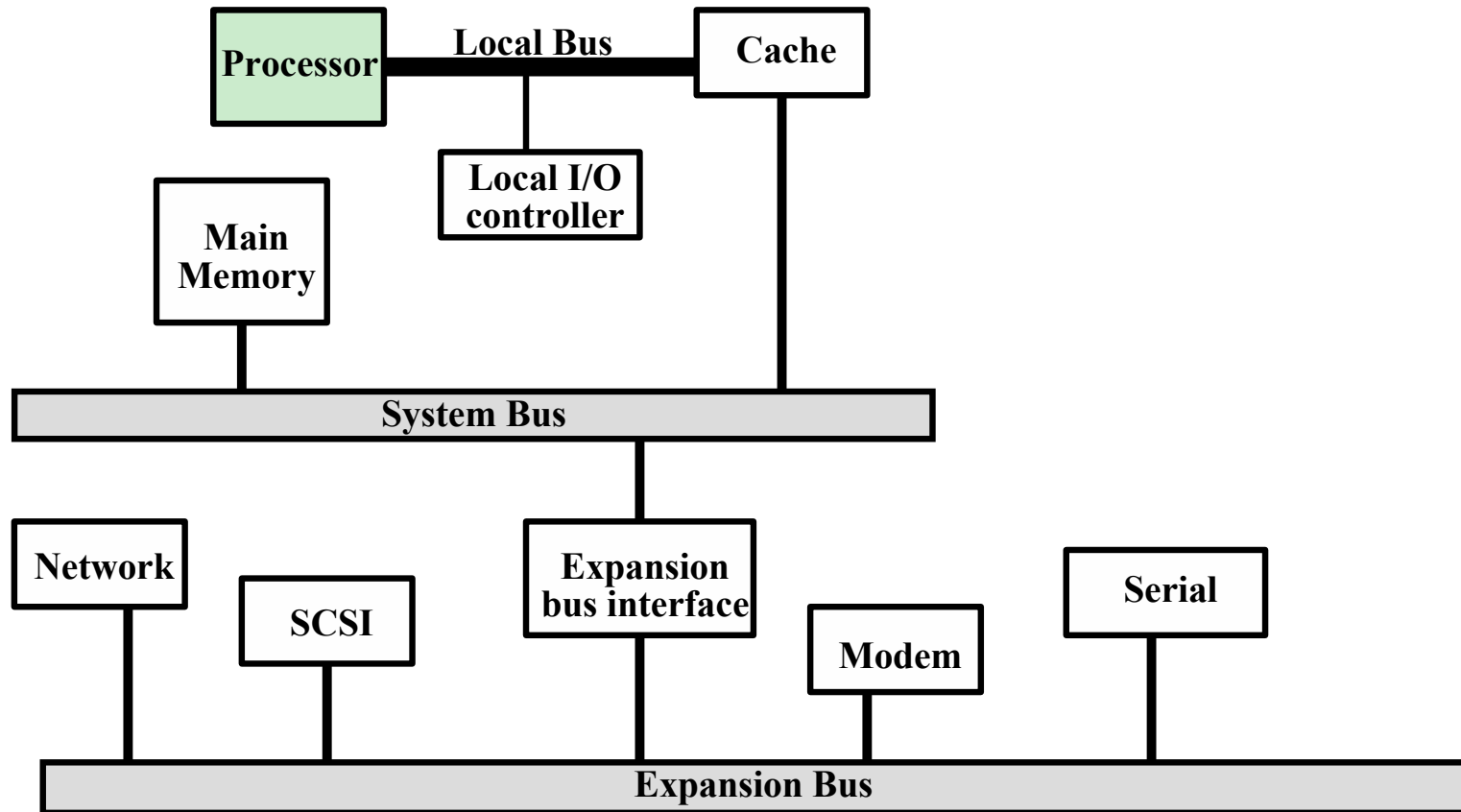
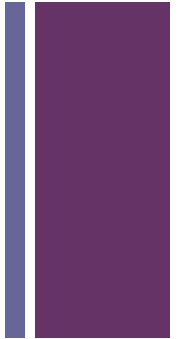


## b. Cấu trúc bus phân cấp



- Nếu một số lượng lớn các thiết bị được kết nối với bus, hiệu suất sẽ giảm. Hai nguyên nhân chính:
  1. Nhiều thiết bị gắn vào bus, chiều dài bus càng dài và do đó trễ truyền càng lớn.
  2. Hiện tượng nút cổ chai: lượng dl cần truyền quá lớn so với khả năng của bus.
- Khắc phục nhược điểm trên: Cấu trúc bus phân cấp
  - **Bus địa phương** kết nối bộ vi xử lý với bus hệ thống qua cache: cách li bộ XL và bộ nhớ → thực hiện DMA
  - **Bus hệ thống** liên kết tất cả các module bộ nhớ chính
  - **Bus mở rộng** kết nối các thiết bị ngoại vi: cho phép nối đc với nhiều tb ngoại vi hơn nhưng vẫn đảm bảo ko làm ảnh hưởng đến bus hệ thống khi dl truyền từ tb ngoại vi quá lớn

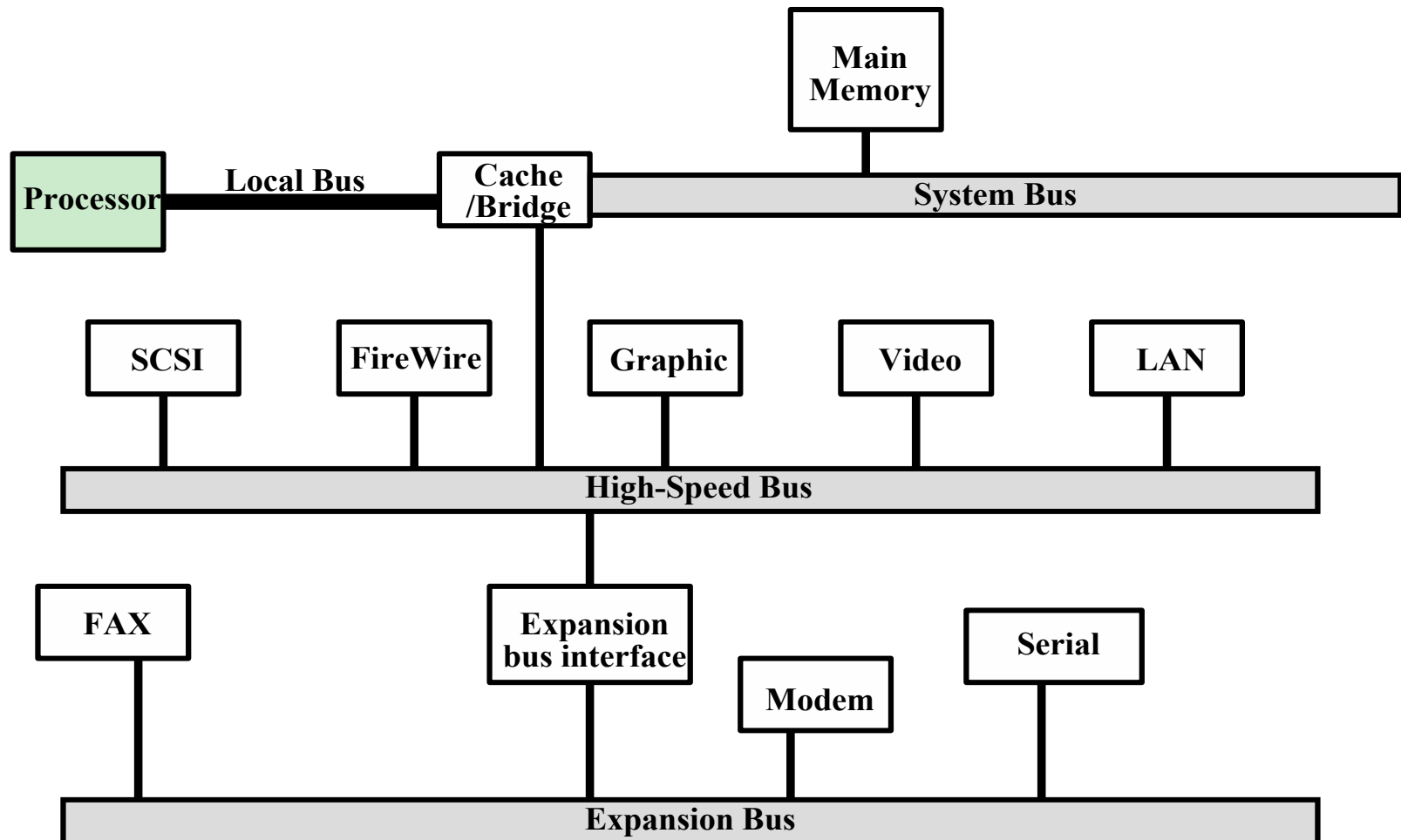
# + Kiến trúc truyền thống (Có cache)



(a) Traditional Bus Architecture



# Kiến trúc hiệu suất cao



(b) High-Performance Architecture

**Figure 3.17 Example Bus Configurations**



## c. Các yếu tố trong thiết kế Bus

### 1. Loại bus

Chuyên dụng

Ghép kênh

### 2. Phương pháp trọng tài

Tập trung

Phân tán

### 3. Định thời

Đồng bộ

Bất đồng bộ

### 4. Độ rộng bus

Địa chỉ

Dữ liệu

### 5. Loại truyền dữ liệu

Đọc

Ghi

Đọc thay đổi ghi

Đọc sau khi ghi

Khối



# 1. Các loại bus: chuyên dụng và ghép kênh.



- Bus chuyên dụng sử dụng cho một chức năng cụ thể: vd: bus dữ liệu, bus địa chỉ, bus điều khiển
  - Ưu điểm: nhanh hơn, ít có xung đột bus
  - Nhược điểm: tăng kích thước và chi phí
- Bus ghép kênh: các thông tin (dữ liệu, địa chỉ) được truyền trên cùng một đường.
  - Sử dụng đường điều khiển AV (Address Valid control line).
  - Khi bắt đầu, đ/c được đưa vào bus và đường AV được kích hoạt.
  - Các module nhận địa chỉ, kiểm tra xem có phải đ/c của nó không.
  - Thông tin đ/c được loại bỏ và một kênh truyền được thiết lập để truyền dữ liệu đọc hoặc ghi
  - Ưu điểm: ít đường hơn, tiết kiệm không gian và chi phí
  - Nhược điểm: mạch phức tạp hơn

## + 2. Phương pháp phân xử (trọng tài)

- Đôi khi, tại một thời điểm có nhiều module cần chiếm bus → cần quyết định xem module nào có quyền sử dụng bus: *phân xử (trọng tài) bus*

→ Phương pháp phân xử bus: tập trung và phân tán

- Phân xử tập trung: bộ điều khiển (bộ phân xử) phân bổ thời gian trên bus. Bộ điều khiển này có thể là một thiết bị riêng hoặc một phần của bộ XL
- Phân xử phân tán: mỗi module chứa một access control logic và chúng làm việc cùng nhau để chia sẻ đường truyền



# 3. Định thời

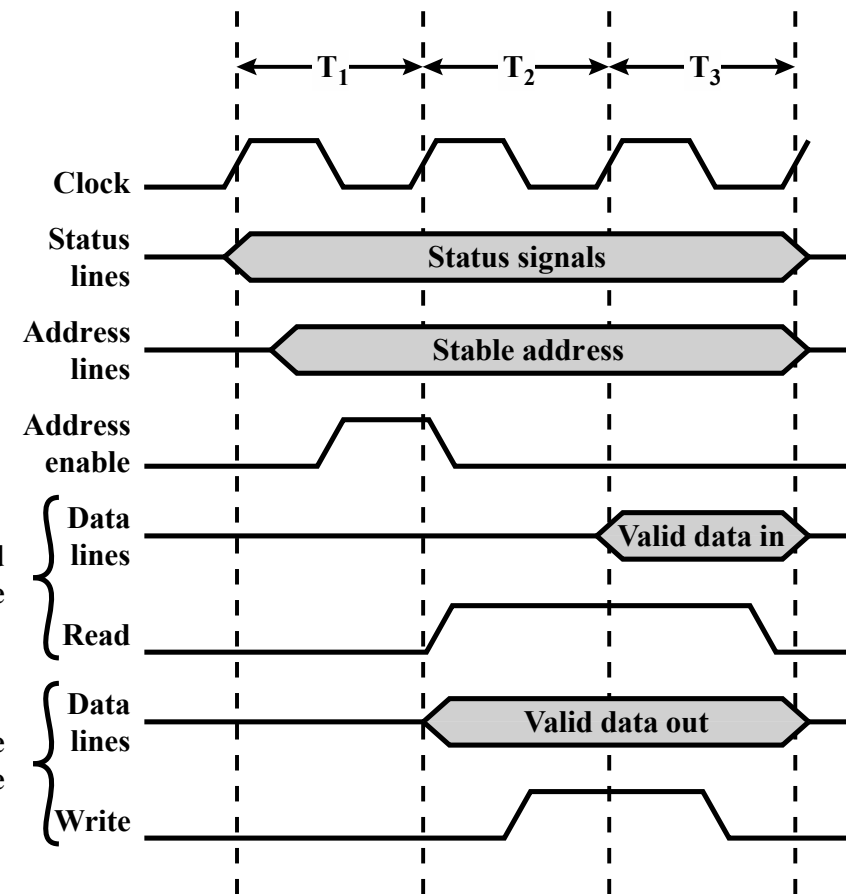


■ **Định thời** là cách các sự kiện được phối hợp truyền trên bus

■ Hai loại: đồng bộ và không đồng bộ

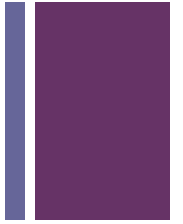
■ Định thời đồng bộ:

- Mỗi hoạt động truyền trên bus được thực hiện theo các xung đồng hồ
- Bus chứa một đường xung đồng hồ (clock line) truyền liên tiếp một chuỗi các bit 0, 1
- Khoảng thời gian T được gọi là chu kỳ đồng hồ
- Tất cả các thiết bị trong máy tính đều đọc được và đồng bộ các hoạt động truyền theo xung này





## 3. Định thời (tiếp)



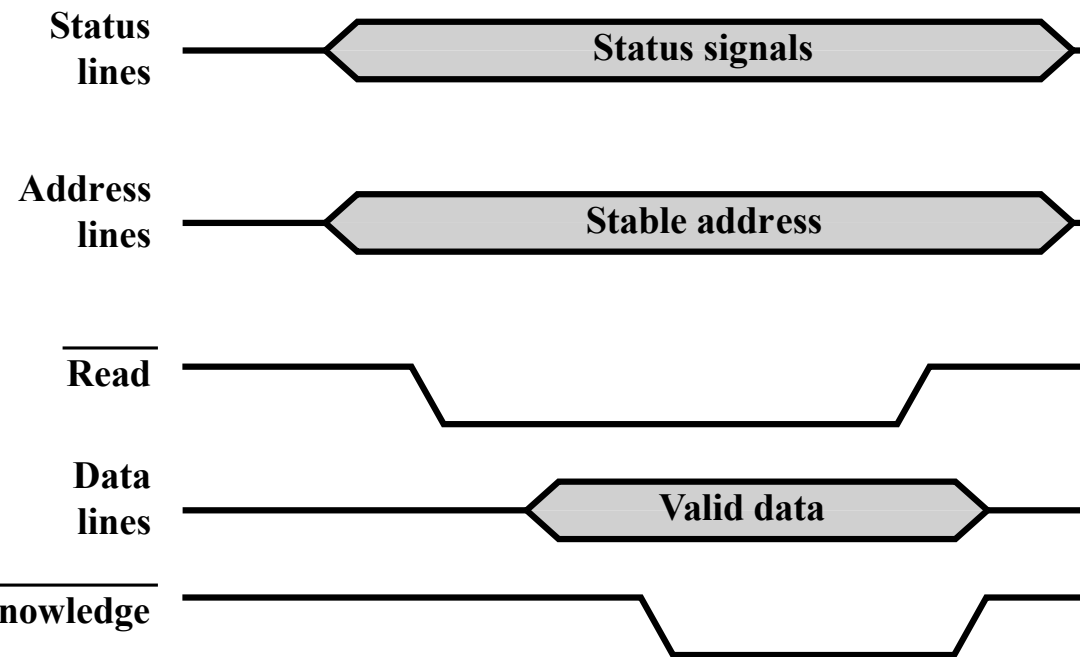
### ■ Định thời không đồng bộ

- Không sử dụng tín hiệu đồng hồ.
- Sau khi dữ liệu được đưa vào bus, bộ nhớ gửi một tín hiệu ACK để báo cho VXL biết việc đọc hoặc ghi dữ liệu

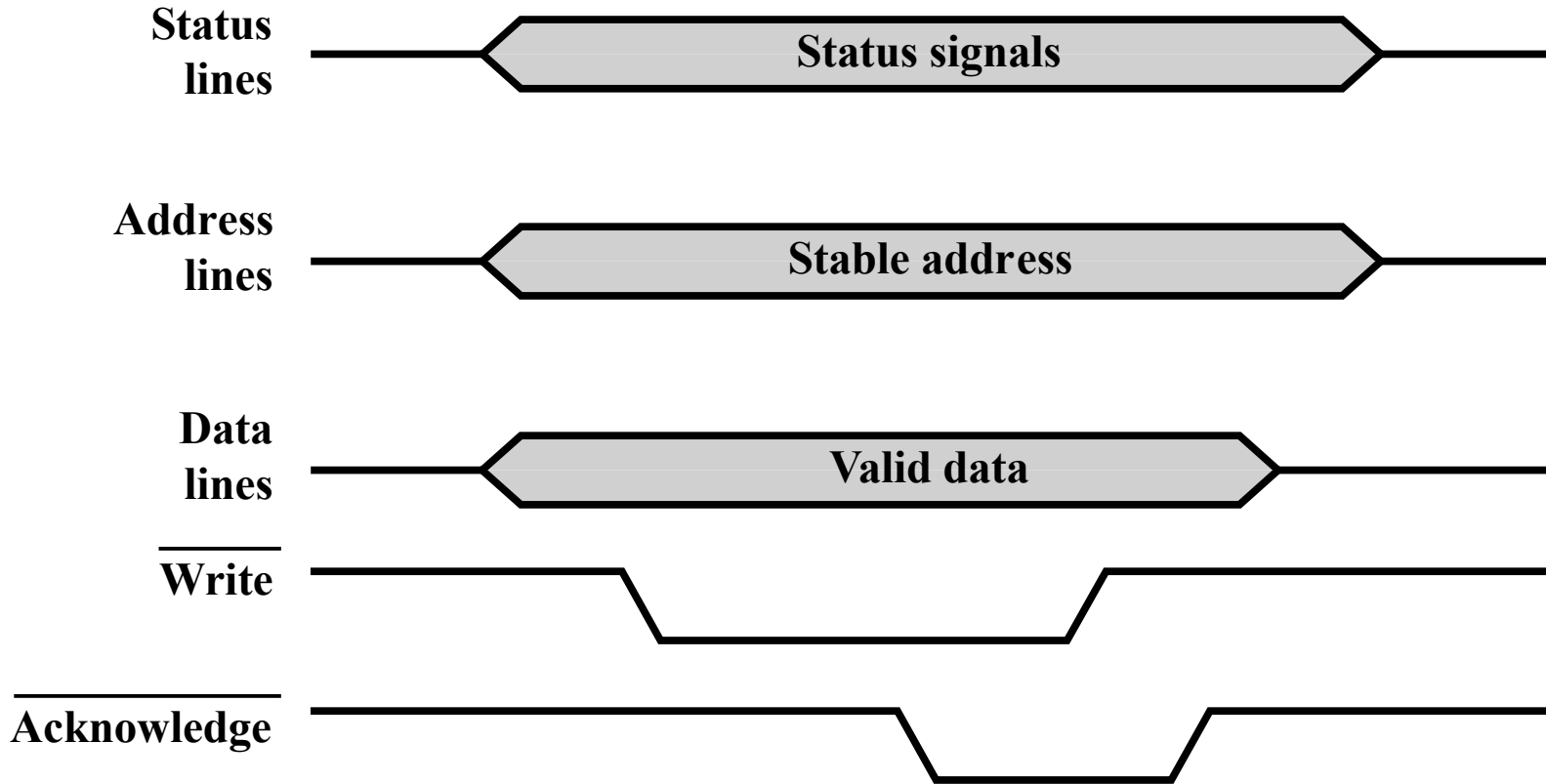
■ Truyền đồng bộ: thực hiện đơn giản tuy nhiên ít linh hoạt hơn truyền không đồng bộ

■ Việc truyền theo xung đồng hồ đôi khi làm giảm hiệu suất hệ thống

■ Truyền không đồng bộ: hiệu quả hơn trong trường hợp nhiều thiết bị có tốc độ xử lý khác nhau chia sẻ chung bus

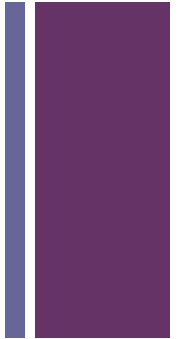


(a) System bus read cycle



(b) System bus write cycle

# + Bài tập



1. Xét một hệ thống máy tính có độ rộng bus địa chỉ là 16b, bus dữ liệu là 16b.
  - a. Không gian địa chỉ bộ nhớ là bao nhiêu?
  - b. Dung lượng tối đa của bộ nhớ là bao nhiêu nếu kích thước mỗi ngăn nhớ là 16b
2. Xét VXL 32b, với bus dữ liệu có độ rộng 16b, hoạt động với tốc độ đồng hồ 8MHz. Giả sử một chu kỳ bus = 4 chu kỳ đồng hồ. Tính tốc độ dữ liệu tối đa được truyền qua bus (theo B/s). Nếu tăng bus dữ liệu lên 32 đường, liệu hiệu suất của hệ thống có được cải thiện?



## + 3.5. Kết nối điểm - điểm

### ■ Nhược điểm của hệ thống kết nối bus:

- Tốc độ của bus đồng bộ không cao do khó khăn trong việc tăng tần số tín hiệu đồng hồ.
- Khi tốc độ dữ liệu cao, việc thực hiện các chức năng đồng bộ và phân xử bus một cách kịp thời trở nên khó khăn hơn
- Với chip đa nhân, nếu sử dụng bus để kết nối, trao đổi dữ liệu giữa các nhân sẽ không đáp ứng được tốc độ VXL → giảm hiệu suất.

### ■ Giải pháp: **kết nối điểm-điểm**: có độ trễ thấp, tốc độ dữ liệu cao, và khả năng mở rộng tốt hơn.

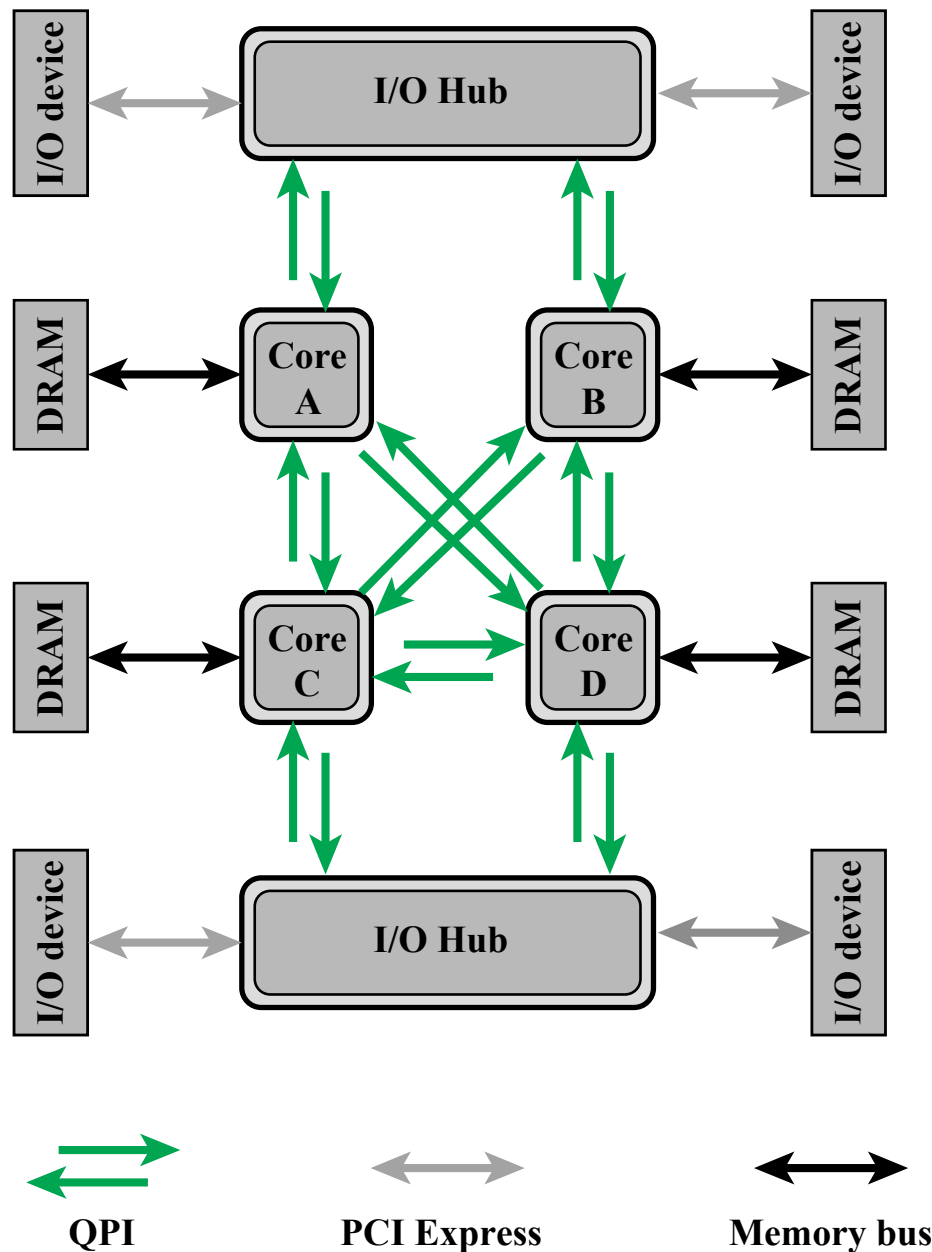
### ■ 2 loại kết nối điểm – điểm: QPI và PCIe

# + Đường dẫn nhanh (Quick Path Interconnect)

- Được giới thiệu vào năm 2008
- Nhiều kết nối trực tiếp
  - Các kết nối từng cặp trực tiếp tới các thành phần khác giúp loại bỏ việc phân xử thường thấy trong các hệ thống truyền dẫn chia sẻ.
- Kiến trúc giao thức phân lớp
  - Những kết nối của bộ xử lý sử dụng kiến trúc giao thức phân lớp chứ không chỉ đơn giản sử dụng tín hiệu điều khiển thường thấy trong sắp xếp các bus chia sẻ.
- Truyền dữ liệu gói
  - Dữ liệu được gửi thành 1 chuỗi các gói chứa tiêu đề điều khiển (header) và mã kiểm soát lỗi.

QPI





Cấu hình  
chip đa  
nhân sử  
dụng  
QPI

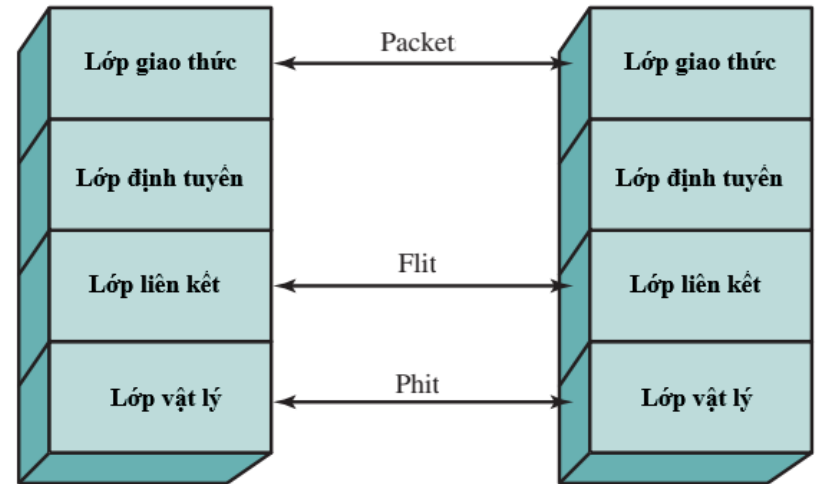
**Figure 3.20 Multicore Configuration Using QPI**



# Các lớp QPI

- QPI được định nghĩa là một kiến trúc bốn lớp, bao gồm các lớp sau:

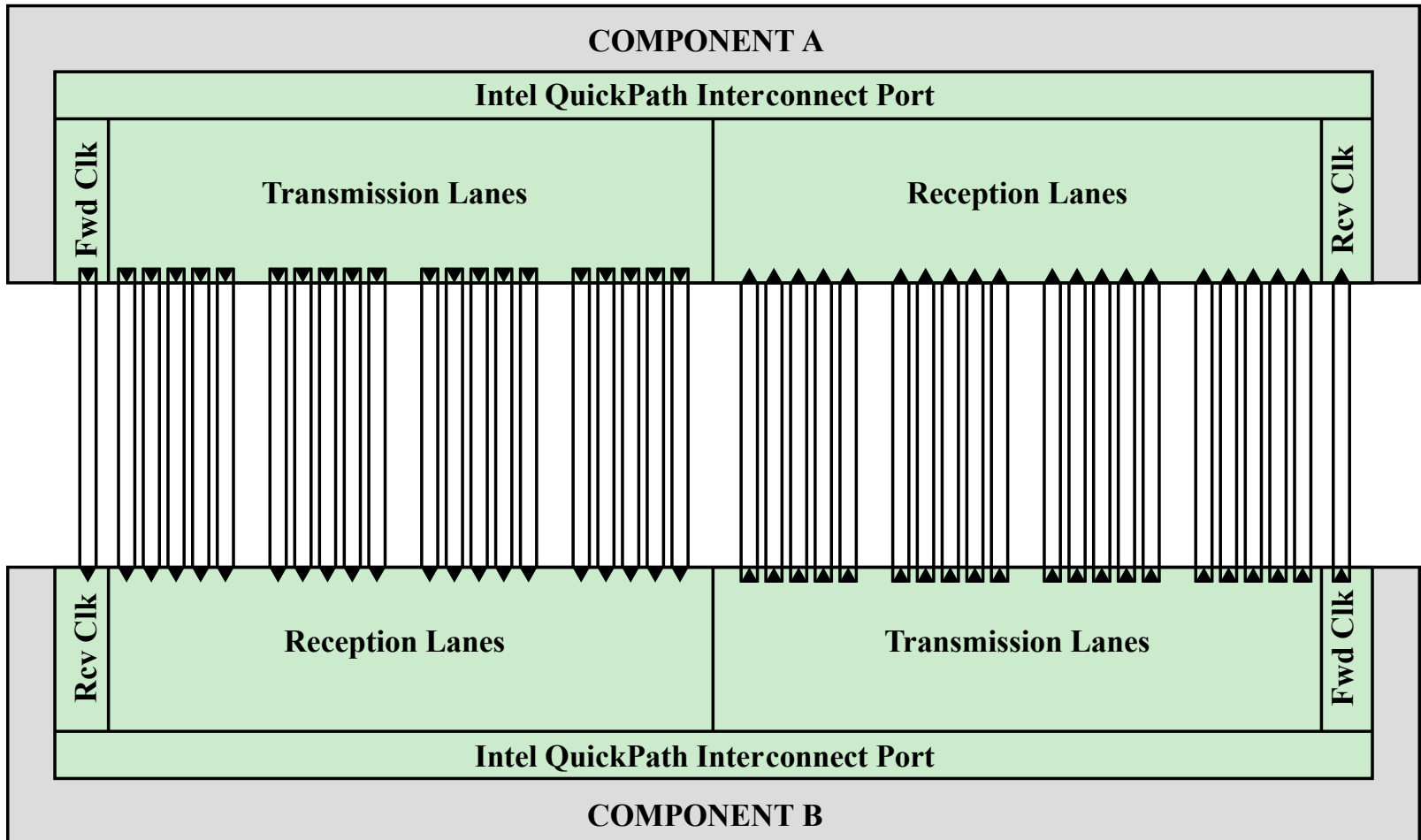
- Vật lý: Bao gồm dây dẫn mang tín hiệu, cũng như mạch và logic để hỗ trợ các tính năng truyền và nhận các bit 1 và 0. Đơn vị chuyển giao ở lớp này 20 bit, được gọi là Phit (physical unit).



- Liên kết: Chịu trách nhiệm truyền tin cậy và điều khiển luồng. Đơn vị dữ liệu của lớp Liên kết là một Flit 80-bit (flow control unit)
- Định tuyến: Cung cấp một framework để chuyển các gói dữ liệu
- Giao thức: Bộ quy tắc để trao đổi các gói tin dữ liệu giữa các thiết bị. Một gói bao gồm một số không đổi các Flit.

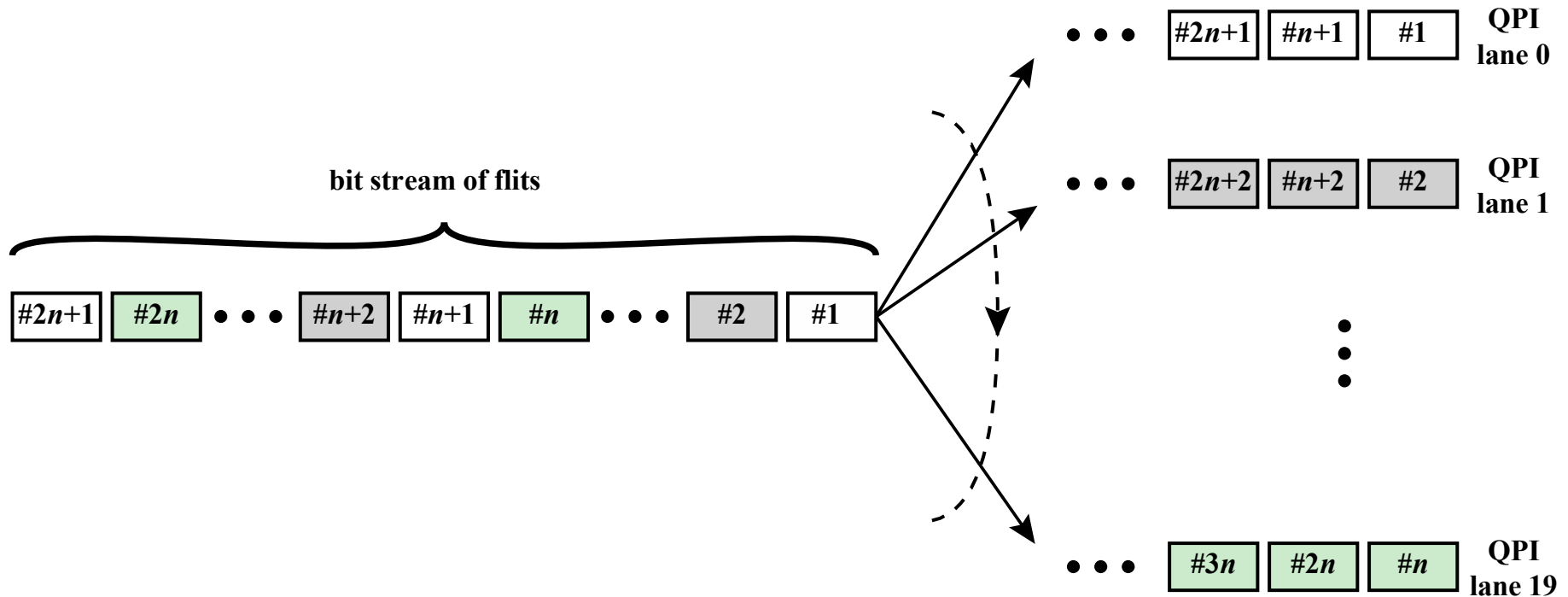
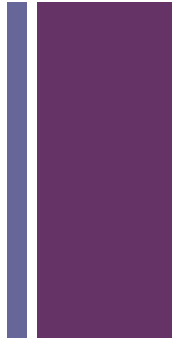
+

## a. Lớp vật lý



+

## b. Lớp liên kết



**Figure 3.23 QPI Multilane Distribution**

## + b. Lớp liên kết (tiếp)

- Thực hiện hai chức năng chính: *điều khiển luồng* và *điều khiển lỗi*.

- Vận hành trên cấp flit (flow control unit – đơn vị điều khiển luồng)
- Mỗi flit gồm 1 bản tin 72-bit và một mã kiểm soát lỗi 8-bit được gọi là *cyclic redundancy check* (CRC)

- Chức năng điều khiển luồng

- Cần thiết để đảm bảo rằng 1 thực thể QPI gửi không áp đảo 1 thực thể QPI nhận bằng cách gửi dữ liệu nhanh hơn khả năng xử lý dữ liệu và xoá bộ đệm để nhiều dữ liệu mới đến của phía nhận

- Chức năng điều khiển lỗi

- Phát hiện và khắc phục lỗi bit, do đó tránh cho các lớp cao hơn gặp lỗi bit



## c. Lớp giao thức và lớp định tuyến

### Lớp Định tuyến

- Được sử dụng để xác định đường đi mà một gói sẽ đi qua các kết nối hệ thống có sẵn
- Xác định bởi phần sụn và mô tả các đường dẫn mà một gói tin có thể đi theo

### Lớp Giao thức

- Gói (packet) là đơn vị truyền
- Một chức năng quan trọng được thực hiện ở lớp này là giao thức liên kết bộ nhớ cache - đảm bảo rằng các giá trị bộ nhớ chính được giữ trong nhiều cache là phù hợp
- Một gói dữ liệu thông thường là một khối dữ liệu được gửi đến hoặc từ một bộ nhớ cache





## 3.6. Kết nối thiết bị ngoại vi

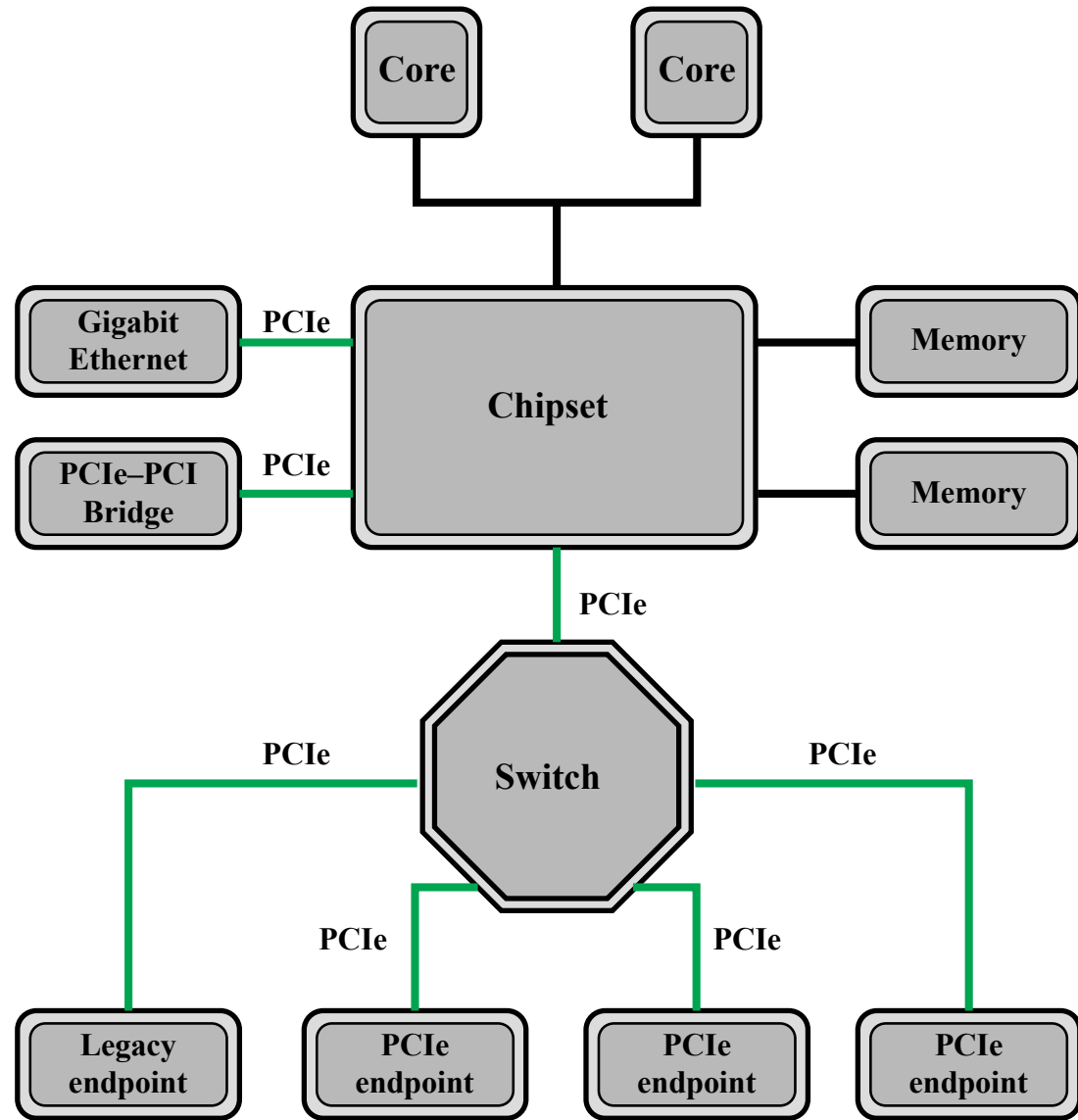
### Peripheral Component Interconnect (PCI)



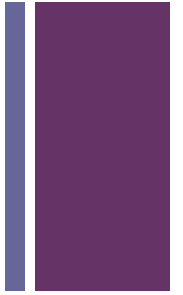
- Một bus băng thông cao, độc lập với bộ xử lý, có thể hoạt động như một bus ngoại vi
- PCI cung cấp hiệu suất cao hơn cho các thiết bị I/O tốc độ cao (vd: card mạng, card màn hình, card ổ cứng)
- Nhóm quan tâm đặc biệt PCI (Special Interest Group - SIG)
  - Được tạo ra để phát triển và duy trì tính tương thích của các đặc tính PCI
- PCI Express (PCIe)
  - Cơ chế kết nối điểm-điểm nhằm thay thế cơ chế dựa trên bus như PCI
  - Yêu cầu chính là dung lượng cao để hỗ trợ nhu cầu của thiết bị I / O tốc độ dữ liệu cao hơn, như Gigabit Ethernet
  - Một yêu cầu khác là phải hỗ trợ các ứng dụng với luồng dữ liệu thời gian thực



# Cấu hình PCIe

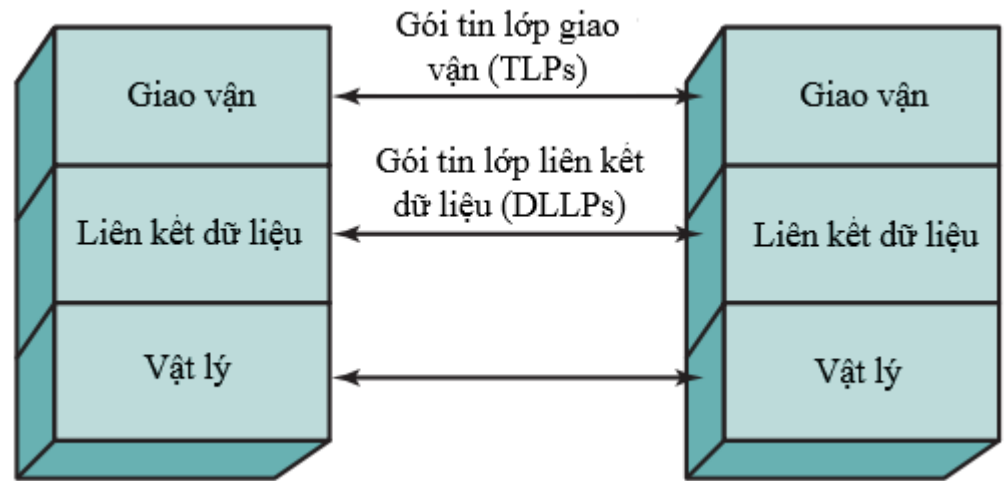


# + Lớp giao thức PCIe



## ■ Kiến trúc giao thức PCIe bao gồm các lớp sau:

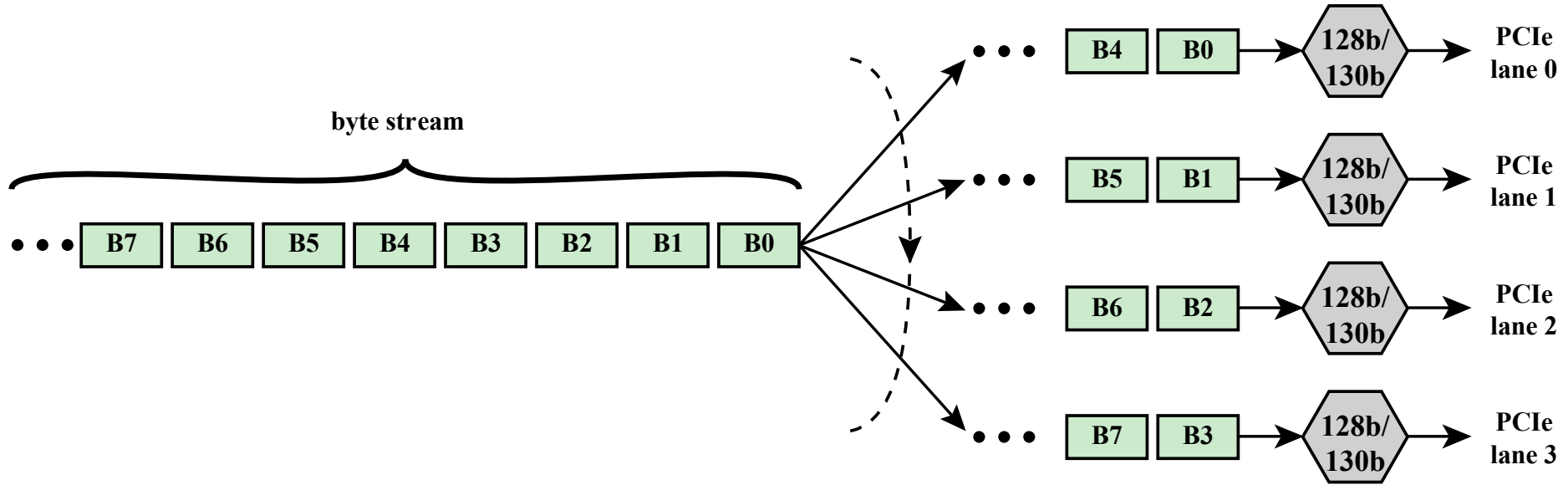
- a. **Vật lý (Physical):** Bao gồm các dây dẫn thực tế mang tín hiệu, cũng như mạch và logic để hỗ trợ các tính năng cần thiết trong việc truyền và nhận các bit 1 và 0.
- b. **Liên kết dữ liệu (Data link layer – DLL):** Chịu trách nhiệm truyền tin cậy và điều khiển luồng. Các gói dữ liệu được tạo ra và được xử lý bởi DLL được gọi là gói DLLP.
- c. **Giao dịch (Transaction Layer):** Tạo ra và xử lý các gói dữ liệu được sử dụng để thực hiện các cơ chế truyền dữ liệu được tải/lưu trữ và cũng quản lý điều khiển luồng của các gói tin giữa hai thiết bị. Các gói dữ liệu của lớp này được gọi là gói TLP.



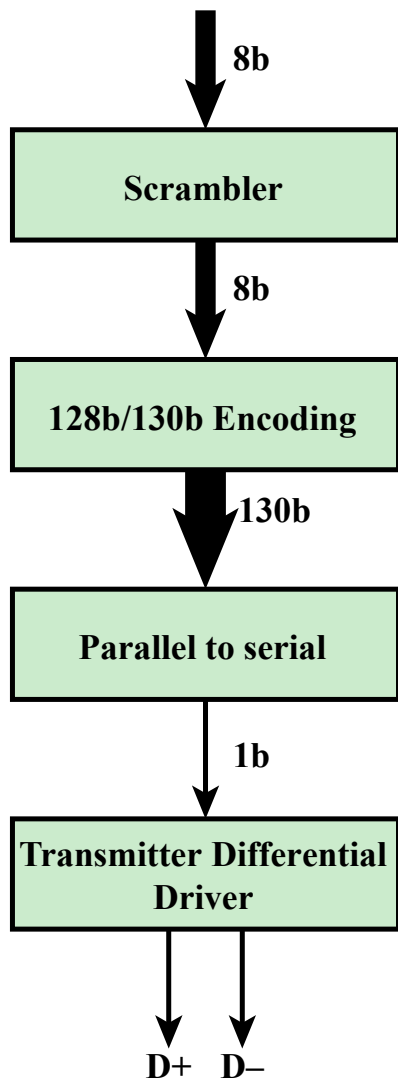
+

# a. Lớp vật lý

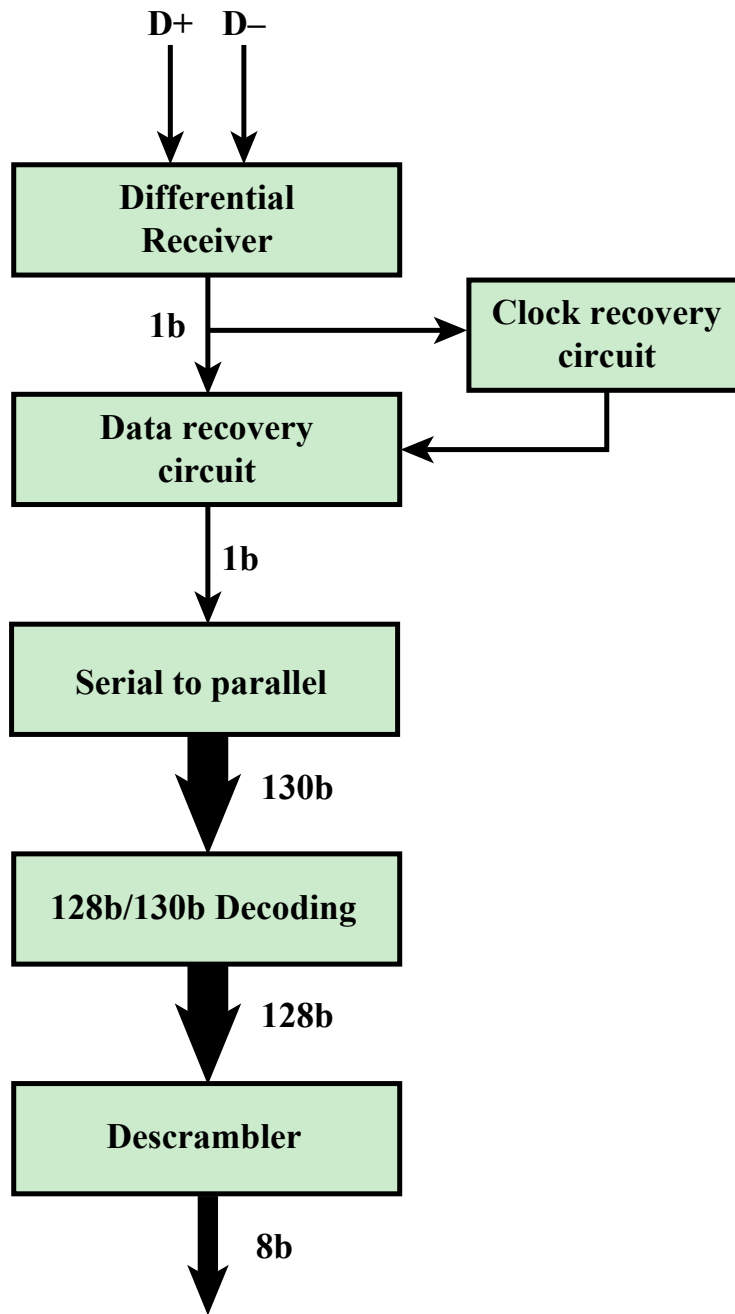
## Phân phối đa tuyến PCIe



**Figure 3.26 PCIe Multilane Distribution**



(a) Transmitter



(b) Receiver

Sơ đồ khối  
Truyền và  
nhận PCIe



## b. Lớp giao dịch PCIe

Transaction Layer (TL)



- Nhận các yêu cầu đọc và ghi từ phần mềm phía trên TL và tạo ra các gói tin yêu cầu để truyền tới đích qua lớp liên kết (link layer)
- Hầu hết các giao dịch sử dụng kỹ thuật giao dịch phân chia
  - Một thiết bị PCIe nguồn gửi 1 gói tin yêu cầu đi, sau đó đợi 1 phản hồi gọi là *gói hoàn thành* (*completion packet*)
- Bản tin TL và một số giao dịch ghi là giao dịch gửi (nghĩa là không cần có phản hồi)
- Định dạng gói TL hỗ trợ địa chỉ bộ nhớ 32-bit và địa chỉ bộ nhớ 64-bit mở rộng

# + TL hỗ trợ bốn không gian địa chỉ:

## ■ Bộ nhớ

- Không gian bộ nhớ bao gồm bộ nhớ chính của hệ thống và thiết bị I/O PCIe
- Các khoảng địa chỉ bộ nhớ nhất định được ánh xạ vào các thiết bị I/O

## ■ Cấu hình

- Không gian địa chỉ này cho phép TL đọc/ghi các thanh ghi cấu hình kết hợp với các thiết bị I/O

## ■ I/O

- Không gian địa chỉ này được sử dụng cho thiết bị PCI kế thừa, với dải địa chỉ dành riêng dùng để xác định các thiết bị I/O kế thừa

## ■ Message

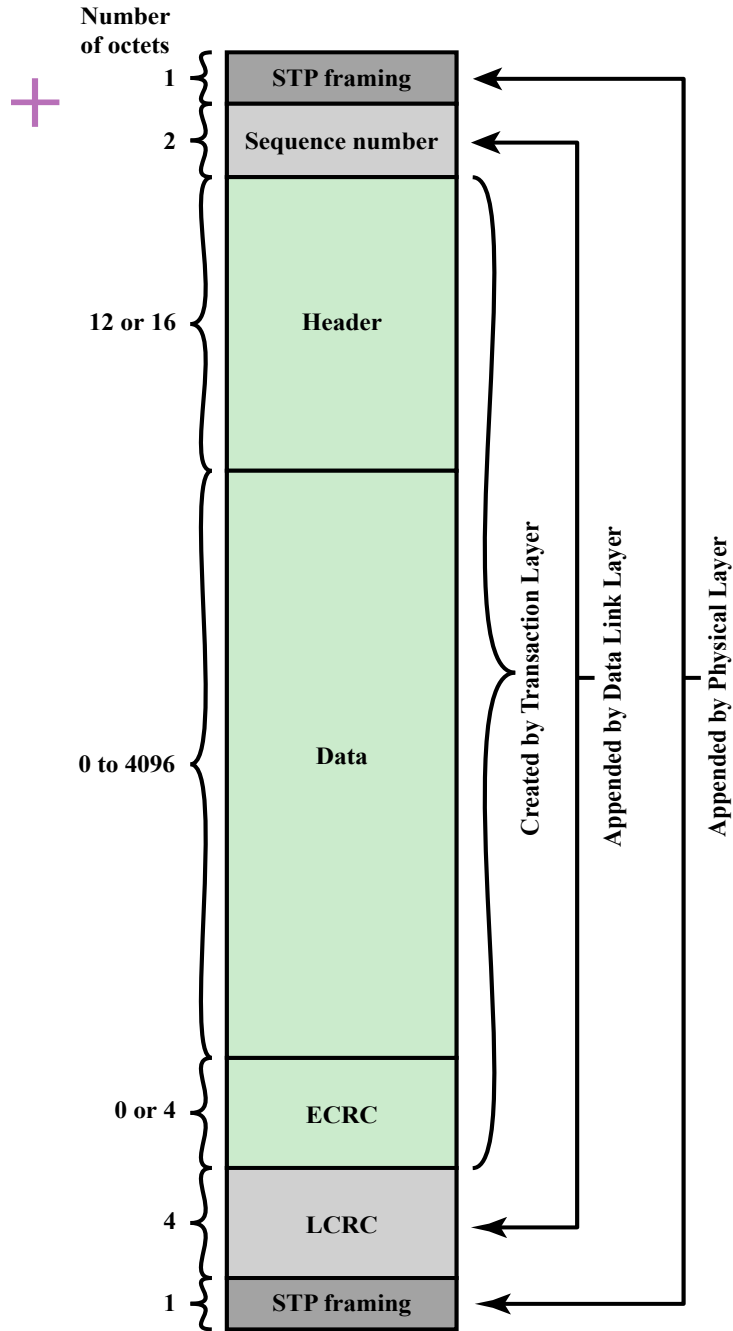
- Không gian địa chỉ này dành cho các tín hiệu điều khiển liên quan đến gián đoạn, xử lý lỗi, và quản lý năng lượng

# Các kiểu giao dịch TLP PCIe

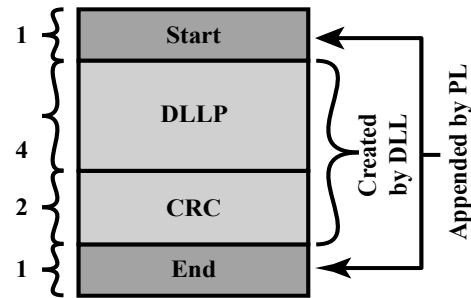


Address Space	TLP Type	Purpose
Memory	Memory Read Request	Transfer data to or from a location in the system memory map.
	Memory Read Lock Request	
	Memory Write Request	
I/O	I/O Read Request	Transfer data to or from a location in the system memory map for legacy devices.
	I/O Write Request	
Configuration	Config Type 0 Read Request	Transfer data to or from a location in the configuration space of a PCIe device.
	Config Type 0 Write Request	
	Config Type 1 Read Request	
	Config Type 1 Write Request	
Message	Message Request	Provides in-band messaging and event reporting.
	Message Request with Data	
Memory, I/O, Configuration	Completion	Returned for certain requests.
	Completion with Data	
	Completion Locked	
	Completion Locked with Data	





(a) Transaction Layer Packet



(b) Data Link Layer Packet

Định dạng  
Đơn vị dữ liệu  
Giao thức PCIe

+

# Định dạng TLP - Yêu cầu bộ nhớ

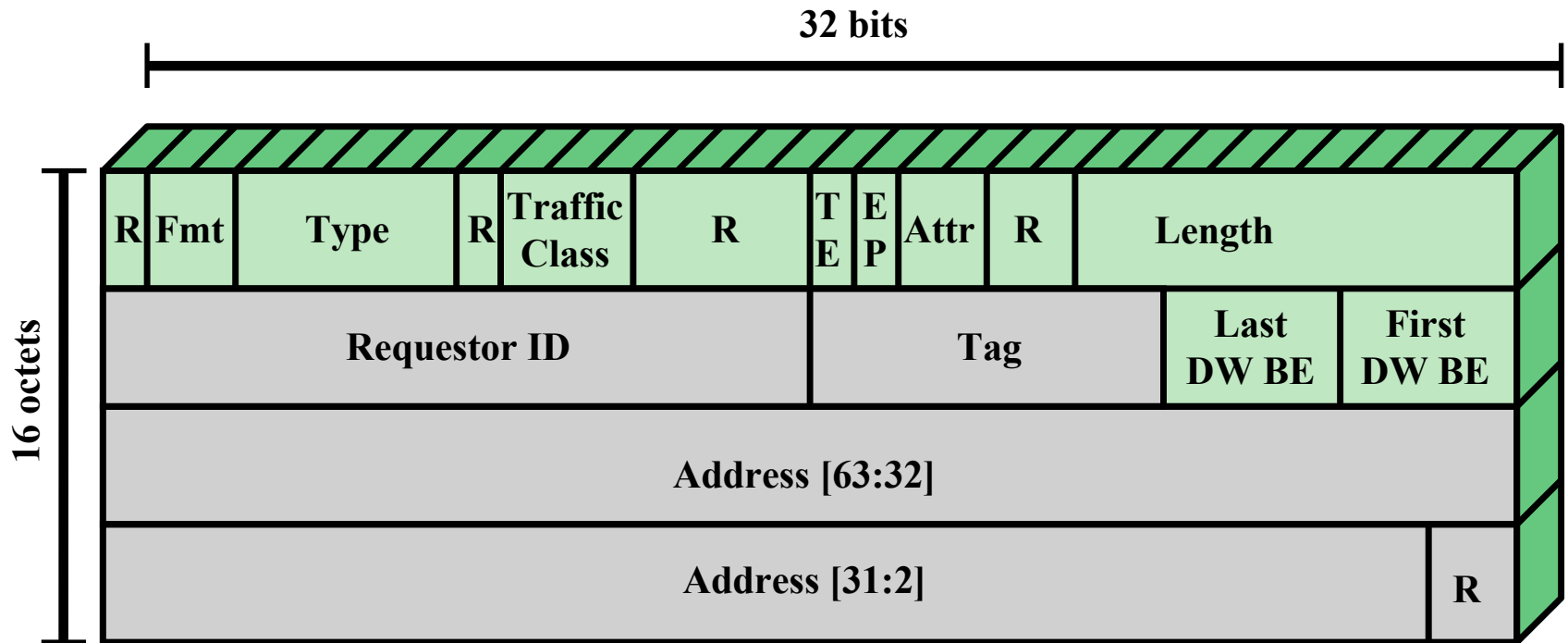


Figure 3.29 TLP Memory Request Format

# + Tổng kết

## Chương 3

## chức năng máy tính và kết nối

- Thành phần máy tính
- Chức năng máy tính
  - Lệnh truy xuất và thi hành
  - Gián đoạn
  - Chức năng I / O
- Cấu trúc kết nối
- Kết nối bus
  - Cấu trúc bus
  - Nhiều phân cấp bus
  - Các yếu tố thiết kế bus

- Kết nối điểm-điểm
  - Lớp vật lý QPI
  - Lớp liên kết QPI
  - Lớp định tuyến QPI
  - Lớp giao thức QPI
- PCI Express
  - Kiến trúc vật lý và logic PCI
  - Lớp vật lý PCIe
  - Lớp giao dịch PCIe
  - Lớp liên kết dữ liệu PCIe



# Câu hỏi

1. Nêu các nhóm chức năng chính của một hệ thống máy tính
2. Các bước thực hiện một lệnh được diễn ra như thế nào?
3. Hai phương pháp xử lý đa ngắt là gì?
4. Các loại tín hiệu truyền nào cần được hệ thống bus hỗ trợ?
5. Các ưu điểm của kiến trúc đa bus so với kiến trúc đơn bus là gì?
6. Các đặc điểm của kết nối điểm – điểm là gì?
7. Liệt kê các lớp của QPI.
8. Liệt kê các lớp của PCIe.

