

การจัดกลุ่มหน่วยประมวลผลแบบ เอ็กทรา-จีซีเอส เพื่อการสื่อสารแบบเพอร์ซันนัลไลซ์ ออล-ทู-ออล บนเครือข่ายการเชื่อมต่อไฮเปอร์คิวบ์แบบชั้น

A Grouping of Processors with Extra-GCS Combination for Parallel ATAPE on Hierarchical Hypercube Systems

^{1*} นันทิพัฒน์ พิศุทธางกูร และ ^{2#} เฉลิมขวัญ สิริพันธุ์

^{1,2}ภาควิชาคอมพิวเตอร์ กองวิชาคณิตศาสตร์และคอมพิวเตอร์ กองการศึกษา โรงเรียนนายเรืออากาศนวมินทกษัตริยาธิราช

^{1*} Nuntipat Phisutthangkoon and ^{2#} Chalermkwan Siripanth

^{1,2}Department of Computer Science, Navaminda Kasatriyadhiraj Royal Air Force Academy

*nuntipat_phis@rtaf.mi.th, #chalermkwan_s@rtaf.mi.th

Received : August 17, 2022

Revised : October 19, 2022

Accepted : December 20, 2022

บทคัดย่อ

เครือข่ายไฮเปอร์คิวบ์แบบชั้น (Hierarchical Hypercube Systems : HHCs) เป็นหนึ่งในเครือข่ายที่ได้รับความนิยมอย่างมากในระบบประมวลผลสมรรถนะสูง (High Performance Computing) เนื่องจากสามารถขยายขนาดได้ง่าย อีกทั้งมีราคาในการสร้างที่ถูกกว่าเครือข่ายไฮเปอร์คิวบ์ (Hypercube Systems : HCs) แบบปกติ อย่างไรก็ตาม สำหรับการสื่อสารเพื่อแลกเปลี่ยนข้อมูลแบบขนานบน HHCs ยังมีความซับซ้อน และอาจจะมีผลกระทบของข้อมูล (Data Collision) เนื่องจาก HHCs มีการลดเส้นเชื่อมระหว่างหน่วยประมวลผลลง ดังนั้น หากปัญหาดังกล่าวได้รับการแก้ไข จะทำให้ HHCs มีความเหมาะสมอย่างมากในการนำไปสร้างลงบนชิปของระบบมัลติโพรเซสเซอร์ (Multiprocessors) ที่รองรับการประมวลผลแบบขนาน เช่น การแลกเปลี่ยนข้อมูลแบบเพอร์ซันนัลไลซ์ ออล-ทู-ออล (All-to-All Personalized Exchange: ATAPE) ดังนั้นงานวิจัยนี้ จึงเสนอการจัดกลุ่มเพื่อการสื่อสารระหว่างหน่วยประมวลผลบน HHCs ในสถานะที่เครือข่ายมีหน่วยประมวลผลถูกใช้งานอยู่เป็นจำนวนมาก หรือสถานการณ์ที่มีข้อจำกัด ทำให้มีหน่วยประมวลผลย่อย ๆ ที่ไม่ถูกใช้งาน โดยเรียกการจัดกลุ่มนี้ว่า การจัดกลุ่มแบบเอ็กทรา-จีซีเอส สำหรับความถูกต้องของการสื่อสารจะถูกทดสอบกับการสื่อสารแบบ ATAPE ระหว่าง k ต้นทาง จนถึง k ปลายทาง ตามลำดับ

คำสำคัญ: การรวมกลุ่มคิวบ์ย่อยแบบไขว้, การหาเส้นทางการสื่อสารแบบขนานที่สั้นที่สุด, กลุ่มของคิวบ์แบบไขว้, การแลกเปลี่ยนแบบเพอร์ซันนัลไลซ์ ออล-ทู-ออล, เครือข่ายไฮเปอร์คิวบ์แบบชั้น

Abstract

Due to higher scalability and lower cost compared with normal Hypercube systems (HCs), Hierarchical Hypercube system (HHCs) is one of the popular models for high performance computing machines. However, edges in HHCs are less than those in HCs and communications between processors in HHCs are more complex, which

can cause possible data collisions in parallel computing. Thus, if these issues can be solved, the performance of HHCs will be enhanced and HHCs can be implemented for a multiprocessor system on a chip of embedded systems which supports parallel computing such as All-to-All Personalized Exchange (ATAPE). Nevertheless, in the case of a network with many processors being used or in critical fragments, some processors in the network are not deployed and can be regrouped to do tasks. To solve critical fragments of systems, this paper proposes an extra-GCS combination. As for the validation, the proposed grouping method was verified using ATAPE communication between k sources and k destinations, respectively.

Keywords: Grouping of cross sub-cube (GCS), Parallel shortest-path routing, Grouping of cross dual-cubes (GCD), All-to-all personalized exchange (ATAPE), Hierarchical hypercube networks (HHCs)

1. บทนำ

ในปัจจุบันงานวิจัยเกี่ยวกับเครือข่ายการเชื่อมต่อระหว่างหน่วยประมวลผล (Interconnection Network) มีบทบาทอย่างมากในการพัฒนาเครือข่ายการประมวลผลแบบขนาน (Parallel Computing System) ซึ่งเครือข่ายการเชื่อมต่อที่มีประสิทธิภาพจะทำให้การสื่อสารระหว่างหน่วยประมวลผล (Processors Elements) มีความรวดเร็วในการส่งข้อมูล และมีความทนทานต่อความผิดพลาด (Fault Tolerance) ดังนั้นการออกแบบเครือข่ายเหล่านี้ สิ่งที่ต้องให้ความสำคัญเป็นพิเศษ คือ โครงสร้างของระบบ (System Topology) เนื่องจากโครงสร้างของระบบจะมีอิทธิพลต่อราคา (Cost) และประสิทธิภาพโดยรวมของระบบ

เครือข่ายการเชื่อมต่อมีสองประเภท ประกอบด้วย เครือข่ายการเชื่อมต่อแบบสถิต (Static Interconnection Network) และเครือข่ายการเชื่อมต่อแบบพลวัต (Dynamic Interconnection Network) โดยที่เครือข่ายการเชื่อมต่อแบบสถิต จะไม่สามารถปรับเปลี่ยนโครงสร้างได้ หรือเรียกอีกอย่างว่าเป็นระบบฝังตัว (Embedded System) ซึ่งเหมาะกับการนำไปใช้ประมวลผลกับการประมวลผลเฉพาะทาง เช่น การนำระบบฝังตัวมาประยุกต์ใช้ร่วมกับระบบติดตามอากาศยาน [1] ที่ต้องการประมวลผลข้อมูลในปริมาณมาก ๆ และจำเป็นต้องประมวลผลพร้อมกันในเวลาเดียวกัน จะทำให้ระบบติดตามอากาศยานประมวลผลได้เร็วขึ้น เนื่องจากระบบฝังตัวเป็นระบบที่ถูกออกแบบมาเฉพาะทาง เพื่อใช้สำหรับงานที่เฉพาะเจาะจง ด้วยเหตุนี้เครือข่ายการเชื่อมต่อแบบสถิต จึงเป็นเครือข่ายที่มีราคาถูกกว่าการเชื่อมต่อแบบพลวัตที่จะมีการปรับเปลี่ยนโครงสร้างของการติดต่อสื่อสารได้ ในอดีตที่ผ่านมาเครือข่ายการเชื่อมต่อแบบสถิตได้ถูกเสนอขึ้นพร้อมกับการคิดค้นขั้นตอนวิธีการสื่อสารอย่างมากมาย [2,3,4,5,6,7,8] ซึ่งหนึ่งในเครือข่ายการเชื่อมต่อที่มีประสิทธิภาพและได้รับความนิยมมากที่สุด คือ เครือข่ายไฮเปอร์คิวบ์ (Hypercube Systems : HCs) [9] เนื่องจากเครือข่ายไฮเปอร์คิวบ์มีคุณสมบัติที่น่าสนใจ เช่น มีค่าเส้นผ่านศูนย์กลาง (Diameter) ที่ต่ำ มีประสิทธิภาพในการสื่อสารที่สูง และมีความสมมาตร (Symmetry) แต่ในการสร้างระบบที่ต้องปรับขยายขนาดเพิ่มได้ตามยุคสมัย โดยเฉพาะยุคที่มีข้อมูลมากมายมหาศาล จะพบว่า ข้อเสียของเครือข่ายไฮเปอร์คิวบ์ คือ ขาดความเหมาะสมในการขยายขนาด เนื่องจากเมื่อมีการเพิ่มขนาดของเครือข่ายหรือเพิ่มจำนวนหน่วยประมวลผลให้มากขึ้น เครือข่ายไฮเปอร์คิวบ์จะมีสัดส่วนด้านราคาที่เพิ่มขึ้นตามไปด้วย ซึ่งราคานั้นสามารถวัดได้จากจำนวนเส้นเชื่อมและหน่วยประมวลผลที่มีมากขึ้น

ด้วยเหตุนี้ จึงมีนักวิจัยเสนอเครือข่ายที่มีความยืดหยุ่นในการขยายขนาด เพื่อรองรับการประมวลผลที่ต้องใช้หน่วยประมวลผลที่เพิ่มมากขึ้น แต่มีราคาถูกกว่าเนื่องจากมีจำนวนเส้นเชื่อมระหว่างหน่วยประมวลผลลดลง เครือข่ายนั้น คือ เครือข่ายไฮเปอร์คิวบ์แบบชั้น (Hierarchical Hypercube Networks : HHCs) ที่มีจำนวนหน่วยประมวลผลจำนวน $N = 2^n$ เมื่อ $n = 2^m + m$ และ $m \geq 2$ [10] นอกจากจะมีความยืดหยุ่นแล้วเครือข่ายไฮเปอร์คิวบ์แบบชั้นยังคงคุณสมบัติเด่น ๆ ของเครือข่าย

ไฮเปอร์คิวบ์ไว้อย่างครบถ้วน เช่น มีความสมมาตร ขนาดของเส้นผ่านศูนย์กลาง และ โหนดคิกิริที่เหมาะสม ถึงแม้จะคงคุณสมบัติเด่นของเครือข่ายไฮเปอร์คิวบ์ไว้มาก แต่เครือข่ายไฮเปอร์คิวบ์แบบชั้นก็ยังมีความยืดหยุ่นด้านการหาเส้นทาง (Routing Paths) เพื่อทำการสื่อสารระหว่างหน่วยประมวลผล เนื่องจากการลดจำนวนเส้นเชื่อมระหว่างหน่วยประมวลผล โดยเฉพาะอย่างยิ่งจะเกิดปัญหาบ่อยในกรณีที่มีการสื่อสารพร้อมกันแบบขนาน ดังนั้นหนึ่งในงานวิจัยบนเครือข่ายไฮเปอร์คิวบ์แบบชั้น ก็คือ การจัดกลุ่มและเส้นทางสำหรับการสื่อสารระหว่างหน่วยประมวลผล หรือโหนด โดยเฉพาะอย่างยิ่งเมื่อต้องการประมวลผลแบบขนาน

ในปี พ.ศ. 2550 ได้มีนักวิจัยคิดค้นการสื่อสารแบบขนานบนเครือข่ายไฮเปอร์คิวบ์แบบชั้น [11] โดยเริ่มส่งจาก 1 ต้นทาง (Source) ไปยัง k ปลายทาง (Destination) แต่ยังมีข้อด้อยตรงที่ต้นทางต้องเริ่มที่โหนดลำดับที่ 0 หรือโหนดแรกของ HHCs เท่านั้น ต่อมาได้มีการคิดค้นการส่งข้อมูลแบบกระจาย (Broadcasting) ในปี พ.ศ. 2554 [12] แต่การส่งข้อมูลนั้นยังไม่ใช้เส้นทางที่สั้นที่สุด หลังจากนั้นในปี พ.ศ. 2564 ได้มีการเสนอการแบ่งกลุ่มของหน่วยประมวลผลที่เรียกว่า คู่อัลคิวบ์แบบไขว้ (Grouping of Cross Dual-cube : GCD) [13] จำนวน 2^{m+1} หน่วยประมวลผล เพื่อการสื่อสารแบบ ATAPE โดยจะส่งข้อความแตกต่างกัน k ข้อความไปยังหน่วยประมวลผลอื่นๆทั้งหมดในระบบ ขั้นตอนวิธีนี้ได้ทำการส่งข้อมูลจากต้นทางไปยังปลายทางอย่างถูกต้อง และเส้นทางที่ได้เป็นเส้นทางที่สั้นที่สุด (Shortest Paths) ล่าสุดในปี 2565 ได้มีนักวิจัยคิดค้นการรวมกลุ่มของหน่วยประมวลผล ที่เรียกว่าการรวมกลุ่มคิวบ์ย่อยแบบไขว้ (Grouping of Cross Sub-cube : GCS) [14] โดยเป็นการรวมกลุ่มเพื่อการสื่อสารแบบ ATAPE เช่นกัน งานวิจัยนี้สามารถเพิ่มจำนวนการรวมกลุ่มของหน่วยประมวลผลได้มากขึ้น (จำนวน $k \geq 2^{m+2}$) แต่อย่างไรก็ตามในกรณีที่เครือข่ายมีหน่วยประมวลผลถูกใช้งานอยู่เป็นจำนวนมากหรือภายใต้สถานการณ์ที่มีข้อจำกัด จะทำให้มีกลุ่มของหน่วยประมวลผลย่อย ๆ กระจุกกระจาย ที่ไม่ได้ถูกใช้งาน ซึ่งงานวิจัยที่ผ่านมายังไม่สามารถใช้ประโยชน์จากหน่วยประมวลผลเหล่านี้ได้เต็มที่

ดังนั้นงานวิจัยนี้ จึงมุ่งเน้นไปที่การรวมกลุ่มของหน่วยประมวลผลภายใต้สถานการณ์ที่มีข้อจำกัด ที่เรียกว่าการจัดกลุ่มคิวบ์ย่อยแบบพิเศษ หรือแบบเอ็กตรา-จีซีเอส (Extra-GCS) โดยมีจำนวนของหน่วยประมวลผลขนาด $k \geq 2^{m+1}$ และสามารถประยุกต์ใช้ในสื่อสารเพื่อแลกเปลี่ยนข้อมูลแบบขนาน ATAPE โดยไม่มีการขัดแย้งกันระหว่างสื่อสาร

2. ขอบเขตงานวิจัย

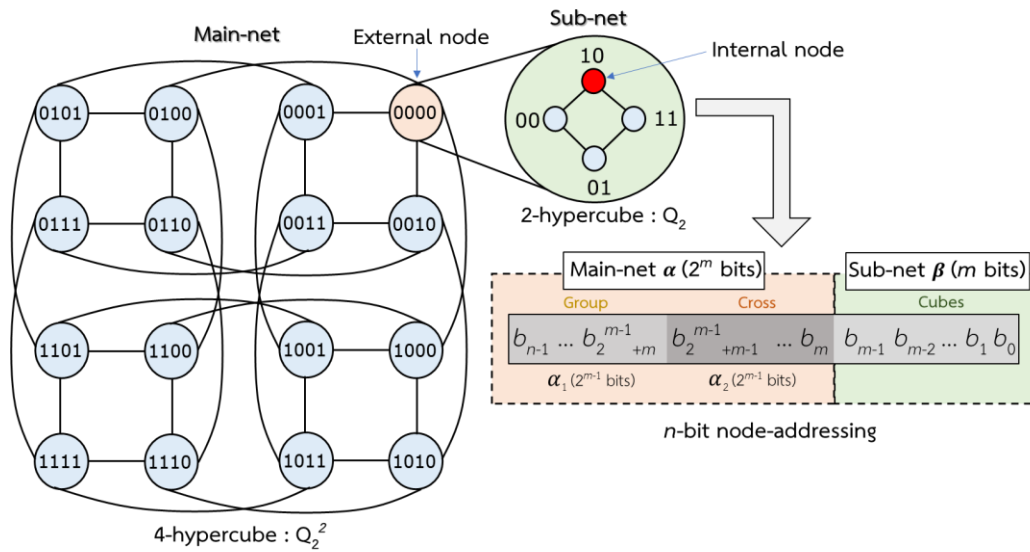
เพื่อจัดกลุ่มของหน่วยประมวลผลบน HHCs ที่รองรับการประมวลผลแบบขนาน ATAPE โดยการจัดกลุ่มนั้นสามารถรองรับความต้องการใช้หน่วยประมวลผลภายใต้สถานการณ์ที่มีข้อจำกัด หรือสถานะที่หน่วยประมวลผลในระบบถูกใช้งานอยู่เป็นจำนวนมากได้

3. ทฤษฎีที่เกี่ยวข้อง

3.1 เครือข่ายการเชื่อมต่อไฮเปอร์คิวบ์แบบชั้น (Hierarchical Hypercube Interconnection Network)

เครือข่ายการเชื่อมต่อไฮเปอร์คิวบ์แบบชั้น [10] จะมีโครงสร้างเป็น 2 ชั้น โดยชั้นแรกสามารถถูกสร้างโดยการนำเครือข่ายไฮเปอร์คิวบ์แบบปกติ (2^m -Hypercube : Q_2^m) มาเป็นโครงสร้างพื้นฐาน และถูกเรียกว่าเครือข่ายหลัก (Main-net) แต่ละโหนด (หรือหน่วยประมวลผล) ในเครือข่ายหลักจะเรียกว่า โหนดชั้นนอก (External Node : α) หลังจากนั้นทำการแทนที่แต่ละโหนด ในเครือข่ายหลักด้วยไฮเปอร์คิวบ์แบบปกติ (m -Hypercube : Q_m) ที่มีขนาดเล็กกว่าอีกครั้งหนึ่ง และเรียกว่าเครือข่ายย่อย หรือคิวบ์ย่อย (Sub-net) และเรียกโหนดในคิวบ์ย่อยว่า โหนดชั้นใน (Internal Node : β) ดังนั้นจำนวนของหน่วยประมวลผลบนเครือข่ายไฮเปอร์คิวบ์แบบชั้น (n -HHCs) จะมีทั้งหมด $N = 2^n$ เมื่อ $n = 2^m + m$ และ $m \geq 2$ ซึ่งแต่ละโหนดสามารถถูกแทนด้วย n -บิต (เลขฐาน 2) ที่ไม่ซ้ำกัน ประกอบด้วย 2^m บิตแรกเป็นของโหนดชั้นนอก และ m บิตหลังเป็นของโหนดชั้นใน ดังแสดงในรูปที่ 1

สำหรับตัวอย่างการอ้างถึงโหนด เช่น ถ้าโหนดชั้นนอก หรือ $\alpha = 0000$ และมีโหนดชั้นใน หรือ $\beta = 01$ นั้น หมายถึง โหนดหมายเลข $0000,01$ (เลขฐาน 2) = โหนดที่ 1 (เลขฐาน 10) (ดูรูปที่ 1 โหนดสีแดง) หรือ โหนดที่มี $\alpha = 0110$ และมี $\beta = 11$ นั้น หมายถึง โหนดหมายเลข $0110,11 =$ โหนดที่ 27



รูปที่ 1 6-HHCs เมื่อ $m = 2$ ขนาด 64 หน่วยประมวลผล และการระบุโหนดที่ระดับบิต

3.2 การแลกเปลี่ยนข้อมูลแบบเพอร์ซันนัลไลซ์ ออล-ทู-ออล (All-to-All Personalized Exchange: ATAPE)

ATAPE คือ หนึ่งในวิธีการสื่อสารในระบบคอมพิวเตอร์แบบขนานและแบบกระจาย (Parallel and Distributed Computers) ที่มีประสิทธิภาพรูปแบบหนึ่ง และถูกนำมาใช้เป็นส่วนหนึ่งของการพัฒนาขั้นตอนวิธีต่าง ๆ ยกตัวอย่างเช่น การกระจายข้อมูลแบบ ออล-ทู-ออล (All-to-all Broadcasting) การทรานสโพสิตริกซ์ (Matrix Transposition) [15] และการแปลงฟูเรียร์ หรือ ฟูเรียร์ทรานสฟอร์มของสัญญาณ (Fast Fourier Transformation) ลักษณะของการสื่อสารแบบ ATAPE คือ การที่ทุกๆ โหนด (จำนวน k โหนด) ทำการส่งข้อความที่แตกต่างกันไปยังโหนดอื่น ๆ ในเครือข่ายทั้งหมด เป็นจำนวน k รอบ (ทำซ้ำ k รอบ) ซึ่งจะใช้ฟังก์ชันเอ็กซลิวซีฟออ์ (XOR-Operation: $\oplus$) [16] ในการสร้างตารางลาติน (Latin Square) เพื่อจัดการส่งข้อมูลจากต้นทาง S (Source) ไปยังปลายทาง D (Destination) ที่ถูกควบคุมด้วยควบคุม C (Control Units) ดังสมการที่ 1 และตัวอย่างตารางลาตินที่ประกอบด้วยหน่วยประมวลผลลำดับที่ 0 - 7 แสดงในรูปที่ 2 สำหรับการอ่านตาราง เช่น ในรอบที่ $C_i = 4$ เมื่อ $0 \leq i \leq k - 1$ จะมีการส่งข้อมูลจากต้นทาง $S = 3$ ไปยังปลายทาง $D = 7$ เป็นต้น

$$D = S \oplus C \quad (1)$$

เมื่อ D = หน่วยประมวลผลปลายทาง
 S = หน่วยประมวลผลต้นทาง
 และ C = ตัวควบคุม

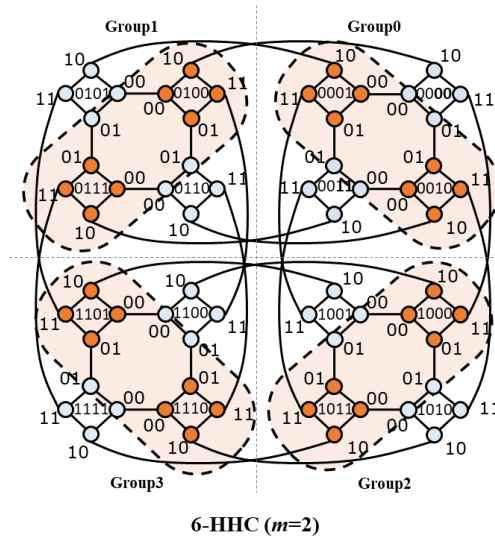
รูปที่ 3 ก) แสดงการจัดกลุ่มแบบคิวบ์ติดกัน (Adjacent Cubes) และ ข) การแบ่งกลุ่มแบบ GCD สังเกตว่า การจัดกลุ่มแบบ Adjacent Cubes จะไม่สามารถสื่อสารแบบ ATAPE ได้ เนื่องจากมีการชนกันในระหว่างการส่งข้อมูลของสองงาน (Task) งานหนึ่งจากต้นทาง S_1 ไปยังปลายทาง D_1 และงานที่สอง จากต้นทาง S_2 ไปยังปลายทาง D_2 ในตำแหน่งเส้นเชื่อม (Edge) $e = (u, v)$ เมื่อ $u = 0000,00$ และ $v = 0001,00$ ดังรูป 2.3 ก) ดังนั้นงานวิจัย [13] จึงเสนอการจัดกลุ่มแบบ GCD มาเพื่อแก้ปัญหการชนกัน โดยจะไปใช้เส้นทางที่สมมาตรกัน (Symmetric Path) และยังได้เสนอการหาเส้นทางที่สมมาตรระดับบิต โดยใช้เทคนิคที่เรียกว่า “การจัดเรียงบิตไปข้างหน้าแบบวนรอบ” (Forward Reordering Bits) และ “การจัดเรียงบิตไปข้างหลังแบบวนรอบ” (Backward Reordering Bits) เพื่อให้สามารถใช้ประโยชน์จากหน่วยประมวลผลในเครือข่ายได้อย่างมีประสิทธิภาพที่สุด โดยขั้นตอนวิธีที่ใช้ในการหาเส้นทางที่สั้นที่สุดแสดงในขั้นตอนวิธีที่ 1

ล่าสุดในปีพ.ศ. 2565 [14] ได้มีการเสนอวิธีการจัดกลุ่มของหน่วยประมวลผลที่ได้จำนวนมากกว่า คือ $2^{m+2} \leq k \leq 2^M$ เมื่อ $M = 2^m - 2^{m-1} + m - 1$ เช่น ถ้า $m = 2$ โดยขั้นตอนการจัดกลุ่มจากงานวิจัยนี้จะได้นหน่วยประมวลผลมากที่สุดทั้งหมดถึง 32 หน่วยประมวลผล หรือครึ่งหนึ่งของจำนวนหน่วยประมวลผลทั้งหมดบน 6-HHCs โดยเรียกวิธีการนี้ว่า “การรวมกลุ่มคิวบ์ย่อยแบบไขว้” (Grouping of Cross Sub-cube : GCS) โดยจะทำการหาบิตของเครือข่ายหลักก่อนเป็นอันดับแรก และทำการหาควอลคิวบ์โหนด (Cross) เป็นลำดับต่อไป การใช้วิธีการแบ่งกลุ่มแบบ GCS พร้อมกับการหาเส้นทางที่สั้นที่สุดด้วยขั้นตอนวิธีที่ 1 จะทำให้สามารถใช้งานหน่วยประมวลผลทั้งหมดในเครือข่ายได้เต็มประสิทธิภาพ 100%

รูปที่ 4 แสดงตัวอย่างสำหรับการแบ่งกลุ่มแบบ GCS บน 6-HHCs ($N = 64, m = 2$) จำนวน 32 หน่วยประมวลผล ประกอบด้วย Main-net = $\{(0001,0010), (0100,0111), (1000,1011), (0001,0010)\}$ หรือประกอบด้วยหน่วยประมวลผลในรูปตัวเลขฐาน 10 ดังต่อไปนี้ $\{4, 5, 6, 7, 8, 9, 10, 11, 16, 17, 18, 19, 28, 29, 30, 31, 32, 33, 34, 35, 44, 45, 46, 47, 52, 53, 54, 55, 56, 57, 58, 59\}$

ขั้นตอนวิธีที่ 1: การหาเส้นทางแบบขนานเพื่อสื่อสารระหว่างหน่วยประมวลผล [13]

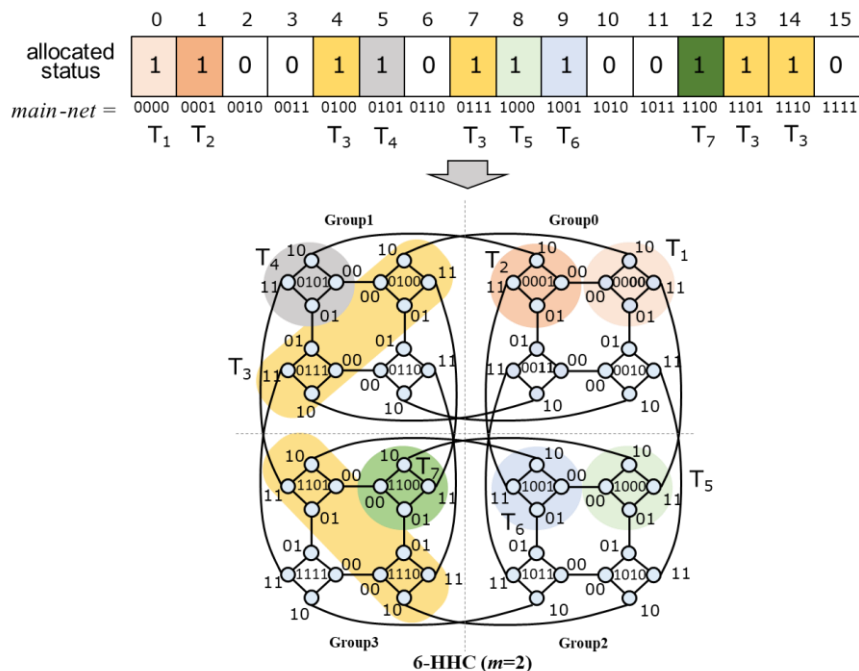
<i>GetParallelSPathHHC</i> 1. $P = \emptyset; \mu = \text{External edge sequence}$ 2. <i>if</i> ($\alpha_s = \alpha_p$) <i>then</i> 3. set P_{in} (HC internal-path routing); 4. <i>else</i> ($\alpha_s \neq \alpha_p$) 5. <i>if</i> ($\beta_s \in \mu$) <i>then</i> $P = P \cup \{ \beta_s, dSPordering(\mu - \{ \beta_s \}) \};$ 6. <i>else</i> $P = P \cup \{ dSPordering(\mu) \};$ 7. <i>return</i> $P;$	
<i>dSPordering</i>(μ) // F: Forward 1. $temp = \beta_s; r' = r;$ 2. $d = (\text{next index of } \beta_s \text{ in } \mu) \% r;$ 3. <i>while</i> ($(p) < r$) <i>do</i> 4. <i>for</i> ($i = 0; i < r'; i++$) <i>do</i> 5. $id = (i + d) \% r';$ 6. $\{ \text{find } \mu[id] \}$ 7. that is closest to $temp;$ 8. <i>end for</i> 9. <i>end while</i> 10. <i>return</i> p	<i>dSPordering</i>(μ) // B: Backward 1. $temp = \beta_s; r' = r;$ 2. $d = (\text{back index of } \beta_s \text{ in } \mu+r) \% r;$ 3. <i>while</i> ($(p) < r$) <i>do</i> 4. <i>for</i> ($i = 0; i < r'; i++$) <i>do</i> 5. $id = (d - i + r') \% r';$ 6. $\{ \text{find } \mu[id] \}$ 7. that is closest to $temp;$ 8. <i>end for</i> 9. <i>end while</i> 10. <i>return</i> p



รูปที่ 4 ตัวอย่างการแบ่งกลุ่มของหน่วยประมวลผลแบบ GCS บน 6-HHCs ($N = 64, m = 2$)
จำนวน 32 หน่วยประมวลผล

4. การดำเนินงานวิจัย

สถานการณ์ที่มีข้อจำกัด คือ สถานการณ์ที่มีกลุ่มของหน่วยประมวลผลกำลังถูกใช้งานอยู่เป็นจำนวนมากในระบบ โดยสามารถตรวจสอบจากสถานะการจัดสรร (Allocated Status) ถ้าเก็บค่า 0 = โหนดว่าง ถ้าเก็บค่า 1 = โหนดไม่ว่าง ซึ่งในสถานการณ์ดังกล่าวอาจจะไม่สามารถจัดสรรหน่วยประมวลผลให้กับงาน (Task) ใหม่ที่มีขนาดใหญ่ได้ ตัวอย่างเช่น รูปที่ 5 แสดง 6-HHCs ($N = 64, m = 2$) ที่มีงาน 7 งาน ที่กำลังใช้หน่วยประมวลผลจำนวน 40 หน่วยประมวลผลอยู่ ประกอบด้วยงาน T_1, T_2, T_4, T_5, T_6 และ T_7 จำนวนงานละ 4 หน่วยประมวลผล อยู่บนโหนดชั้นนอก 0000, 0001, 0101, 1000, 1001 และ 1100 ตามลำดับ และมีงาน T_3 จำนวน 16 หน่วยประมวลผล อยู่บนโหนดชั้นนอกแบบ Cross-node (0100, 0111, 1101, 1110)



รูปที่ 5 การจัดสรรงานลง 6-HHCs จำนวน 7 งาน

สังเกตว่า งานที่มีอยู่บนเครือข่ายจะไม่สามารถใช้ขั้นตอนวิธีการจัดกลุ่มแบบ GCD หรือ GCS ในการรองรับงานใหม่ที่มีขนาดใหญ่ได้ ($k \geq 2^{m+1}$) เช่น ในกรณีที่มีงานใหม่ (Incoming Tasks) ที่ต้องการหน่วยประมวลผล ATAPE จำนวน 16 หน่วยประมวลผลจะไม่สามารถจัดสรรลงในเครือข่าย ณ เวลาปัจจุบันได้ ดังนั้น การรวมกลุ่มแบบเอ็กทรา-จีซีเอส จึงถูกเสนอขึ้นมาเพื่อรองรับสถานการณ์นี้

งานวิจัยนี้จะทำการเลือกหนึ่งโหนดชั้นนอก (External Node) บนแต่ละกลุ่มของโหนดชั้นนอก (External Node Grouping : Group) มารวมกันเป็นเอ็กทรา-จีซีเอส ที่สามารถรวมกลุ่มของหน่วยประมวลผลจำนวน $k = 2^{m+1}$ ถึง 2^M โหนด เมื่อ $M = 2^m - 2^{m-1} + m - 1$ โดยที่ Group_g จะหาได้จาก 2 บิตแรกของโหนดชั้นนอกนับจากซ้ายมือ เมื่อ $0 \leq g \leq 2^m - 1$ เมื่อ $w = 2^{m-1}$ เช่น โหนด 0110 บน 6-HHCs มี 2 บิตแรก คือ 01 นั้นแสดงให้เห็นว่าโหนด 0110 จะอยู่ใน Group_1 เป็นต้น

ขั้นตอนวิธีสำหรับการรวมกลุ่มแบบเอ็กทรา-จีซีเอส เริ่มโดยการเลือกโหนดแรกที่ว่าง (Available) เรียกว่า โหนดอินพุต (Input Node) โดยกำหนดให้ โหนดอินพุตของโหนดชั้นนอกแทนด้วย $A = \alpha_{1A}\alpha_{2A}$ และ โหนดผลลัพธ์ (Output) ของโหนดชั้นนอก แทนด้วย $D = \alpha_{1D}\alpha_{2D}$ โดยที่ โหนดชั้นนอก D จะต้องเป็นโหนดที่อยู่ใน Group ที่อยู่ติดกันกับโหนดชั้นนอก A (Adjacent Group) และขั้นตอนต่อมาจะทำการหาครอส (Cross) ของโหนดชั้นนอก A ดังนั้น กลุ่มที่ติดกันกลุ่มที่ i ของกลุ่ม $\alpha_{1A} = \alpha_{1A} \oplus 2^i$ เมื่อ $0 \leq i \leq w-1$ และ D จะสามารถหาได้โดยใช้สมการที่ 2

$$\begin{aligned} \text{Group } \alpha_{1D} \text{ (ของ } D) &= i \\ \text{Cross } \alpha_{2D} \text{ (ของ } D) &= \overline{\alpha_{2A} \oplus i} \end{aligned} \quad (2)$$

ตารางที่ 1 แสดงตัวอย่างของการรวมกลุ่มแบบเอ็กทรา-จีซีเอส สำหรับหน่วยประมวลผลขนาด $k = 2^{m+1} = 8$ โหนด บน 6-HHCs โดยกำหนดให้ โหนดอินพุต A ประกอบด้วยสี่โหนดดังนี้ 0000, 0001, 0010 และ 0011 สังเกตว่าผลลัพธ์ของโหนดชั้นนอก D ของแต่ละโหนดอินพุต A จะประกอบด้วยสองโหนด คือ D_1 และ D_2 เนื่องจากกลุ่มที่อยู่ติดกันกับกลุ่มของโหนด A ประกอบด้วยสองกลุ่ม เช่น ถ้าโหนดข้อมูลเข้า $A = \alpha_{1A}\alpha_{2A} = 0000$ จะได้ว่ามี $i = 1$ และ 2 เมื่อ $i = 1$ (หรือ $\alpha_{1D} = 01$) จะได้ $\alpha_{2D} = \overline{\alpha_{2A} \oplus i} = \overline{00 \oplus 01} = 10$ ดังนั้นจะได้ $D_1 = \alpha_{1D}\alpha_{2D} = 0110$ ซึ่งจะได้การรวมกลุ่มของเอ็กทรา-จีซีเอส จำนวน 8 โหนด คือ (0000, xx) และ (0110, xx) โดยมีจำนวนหน่วยประมวลผลทั้งหมดคือ (0, 1, 2, 3, 24, 25, 26, 27) และในทำนองเดียวกัน ถ้า $i = 2$ (หรือ 10) จะได้ $D_2 = 1001$ และมีการรวมกลุ่มของเอ็กทรา-จีซีเอส จำนวน 8 โหนด คือ 0000, xx และ 1001, xx และตัวอย่างนี้แสดงในรูปที่ 6 ก) และรูปที่ 6 ข) แสดงตัวอย่างเมื่อ $A = 0011$

อย่างไรก็ตามภายใต้สถานการณ์ที่มีข้อจำกัด และกรณีที่ต้องการหน่วยประมวลผลจำนวน $k > 2^{m+1}$ ถึง 2^M โหนด เมื่อ $M = 2^m - 2^{m-1} + m - 1$ สามารถใช้สมการที่ 2 ร่วมกันกับสมการที่ 3 เพื่อรวมกลุ่ม 4 กลุ่ม (หรือสูงสุด 2^m กลุ่ม) โดยจะใช้เพียงหนึ่งโหนดชั้นนอกต่อหนึ่งกลุ่มเท่านั้น เช่น A, D_1, D_2 และ D_3 เมื่อ $k = 2^{m+2} = 16$ บน 6-HHCs และสามารถหาโหนดชั้นนอกโหนดสุดท้าย (เช่น D_3) ได้จากสมการที่ 3 ดังนี้

$$\begin{aligned} \text{Group } \alpha_{1D} \text{ (ของ } D) &= 11 \dots 1 \\ \text{Cross } \alpha_{2D} \text{ (ของ } D) &= \overline{\alpha_{2A}} \end{aligned} \quad (3)$$

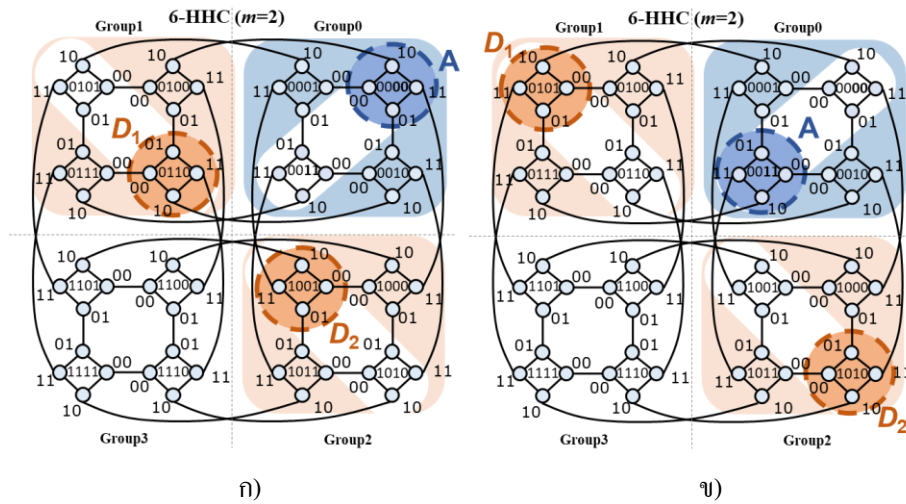
ตัวอย่างเช่น กรณีที่ต้องการหน่วยประมวลผลจำนวน 16 หน่วยประมวลผล โดยมีโหนด $A = 0000$ จะได้ว่า มี $i = 1$ (หรือ 01) และ 2 (หรือ 10) ซึ่งขั้นตอนการหาโหนด D_1 และ D_2 สามารถอธิบายได้ด้วยตัวอย่างก่อนหน้า ซึ่งจะได้ว่า $D_1 = 0110$ และ $D_2 = 1001$ สำหรับการหา D_3 จะใช้สมการที่ 3 จะได้ว่า $\alpha_{1D} = 11$ และ $\alpha_{2D} = \overline{00} = 11$ ดังนั้นจะได้ $D_3 = \alpha_{1D} \alpha_{2D} = 1111$ ดังที่แสดงตัวอย่างในรูปที่ 7 ก) และ $A = 0011$ ในรูปที่ 7 ข) หรือตารางที่ 2

ตารางที่ 1 เอ็กทรา-จีซีเอส สำหรับ $k = 2^{m+1}$ บน 6-HHCs ($m = 2$ และ $A = 0 - 3$)

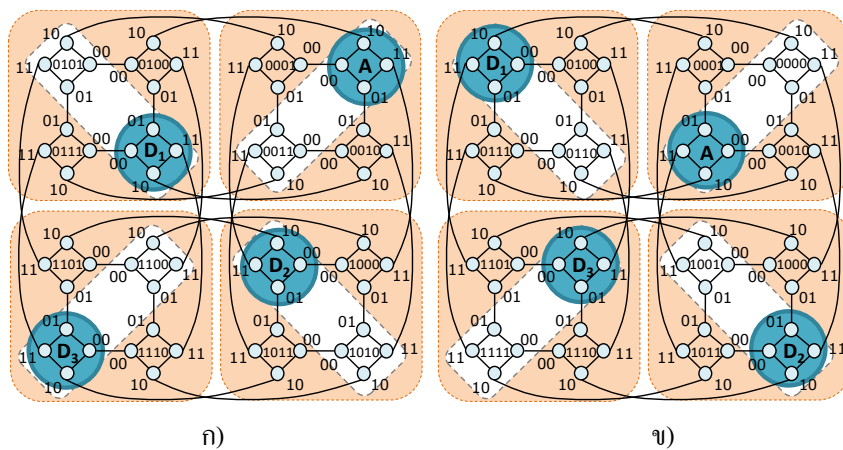
A	α_{1D}	α_{2D}	$D = \alpha_{1D} \alpha_{2D}$	β	$2^{m+1} = 8 \text{ Ps } (4 \times 2)$
0000	01	10	$D_1 = 0110$	00 - 11	<u>0</u> ,1,2,3, <u>24</u> ,25,26,27
0000	10	01	$D_2 = 1001$	00 - 11	<u>0</u> ,1,2,3, <u>36</u> ,37,38,39
0001	01	11	$D_1 = 0111$	00 - 11	<u>4</u> ,5,6,7, <u>28</u> ,29,30,31
0001	10	00	$D_2 = 1000$	00 - 11	<u>4</u> ,5,6,7, <u>32</u> ,33,34,35
0010	01	00	$D_1 = 0100$	00 - 11	<u>8</u> ,9,10,11, <u>16</u> ,17,18,19
0010	10	11	$D_2 = 1011$	00 - 11	<u>8</u> ,9,10,11, <u>44</u> ,45,46,47
0011	01	01	$D_1 = 0101$	00 - 11	<u>12</u> ,13,14,15, <u>20</u> ,21,22,23
0011	10	10	$D_2 = 1010$	00 - 11	<u>12</u> ,13,14,15, <u>40</u> ,41,42,43

ตารางที่ 2 เอ็กทรา-จีซีเอส สำหรับ $k = 2^{m+2}$ บน 6-HHCs ($m = 2$ และ $A = 0 - 3$)

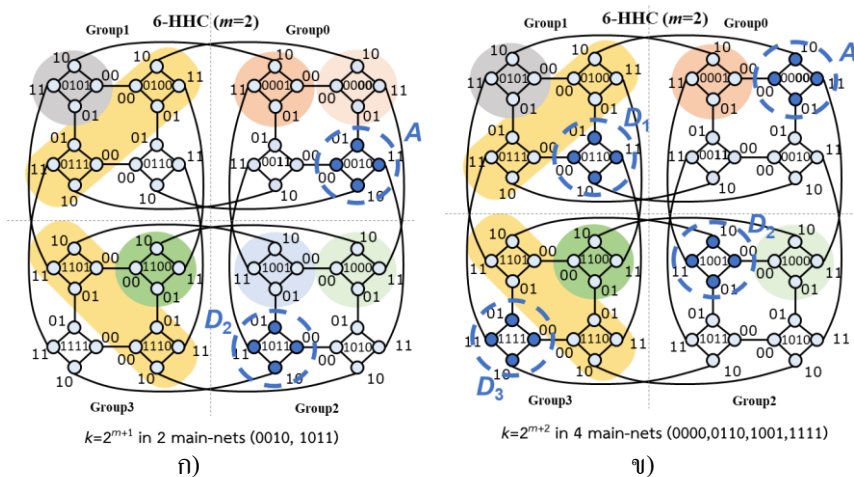
A	α_{1D}	α_{2D}	$D = \alpha_{1D} \alpha_{2D}$	β	$2^{m+1} = 8 \text{ Ps } (4 \times 2)$
0000	00	00	$A = 0000$	00 - 11	0, 1, 2, 3
	01	10	$D_1 = 0110$	00 - 11	24, 25, 26, 27
	10	01	$D_2 = 1001$	00 - 11	36, 37, 38, 39
	11	11	$D_3 = 1111$	00 - 11	60, 61, 62, 63
0001	00	01	$A = 0001$	00 - 11	4, 5, 6, 7
	01	11	$D_1 = 0111$	00 - 11	28, 29, 30, 31
	10	00	$D_2 = 1000$	00 - 11	32, 33, 34, 35
	11	10	$D_3 = 1110$	00 - 11	56, 57, 58, 59
0010	00	10	$A = 0010$	00 - 11	8, 9, 10, 11
	01	00	$D_1 = 0100$	00 - 11	16, 17, 18, 19
	10	11	$D_2 = 1011$	00 - 11	44, 45, 46, 47
	11	01	$D_3 = 1101$	00 - 11	52, 53, 54, 55
0011	00	11	$A = 0011$	00 - 11	12, 13, 14, 15
	01	01	$D_1 = 0101$	00 - 11	20, 21, 22, 23
	10	10	$D_2 = 1010$	00 - 11	40, 41, 42, 43
	11	00	$D_3 = 1100$	00 - 11	48, 49, 50, 51



รูปที่ 6 ตัวอย่างการรวมกลุ่มแบบเอ็กทรา-จีซีเอส เมื่อ $A = 0000$ ก) และ 0011
ข) เครือข่าย 6-HHCs ($N = 64, m = 2$) ที่มีหน่วยประมวลผลขนาด $k = 2^{m+1}$



รูปที่ 7 ตัวอย่างการรวมกลุ่มแบบเอ็กทรา-จีซีเอส เมื่อ $A = 0000$ ก) และ 0011
ข) เครือข่าย 6-HHCs ($N = 64, m = 2$) ที่มีหน่วยประมวลผลขนาด $k = 2^{m+2}$



รูปที่ 8 ตัวอย่างการรวมกลุ่มแบบเอ็กทรา-จีซีเอส ขนาด $k = 2^{m+1}$, $A = 0010$ ก)
และ $k = 2^{m+2}$, $A = 0000$ ข) บน 6-HHCs ($N = 64, m = 2$)

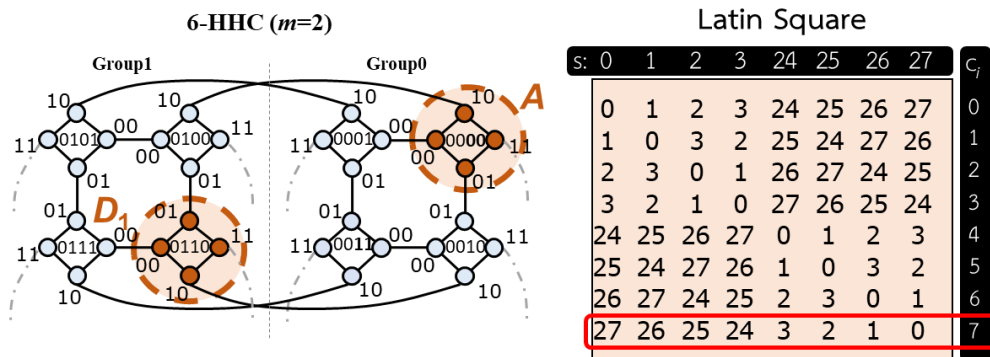
รูปที่ 8 แสดงตัวอย่างสำหรับการรวมกลุ่มภายใต้สถานการณ์ที่มีข้อจำกัดด้วยเอ็กทรา-จีซีเอส สมมติให้ มีโหนดชั้นนอก (External Node) บน 6-HHCs มีหน่วยประมวลผลว่างอยู่ 6 โหนด หรือ 24 หน่วยประมวลผลจากทั้งหมด 16 โหนดชั้นนอก (ดูรูปที่ 8 ก)) ถ้าระบบต้องการหน่วยประมวลผลขนาด $k=2^{m+1}=8$ หน่วยประมวลผลสำหรับการ ATAPE แต่ไม่สามารถรวมกลุ่มของหน่วยประมวลผลแบบ GCD สำหรับงานที่ต้องการได้ ดังนั้นการจัดสรรกลุ่มของ 8 หน่วยประมวลผลนี้จะต้องใช้การรวมกลุ่มแบบเอ็กทรา-จีซีเอส โดยโหนดข้อมูลเข้าโหนดแรก คือ $A = 0010$ และใช้สมการที่ 2 เพื่อหาตัวคูณของ A จะได้ว่า $D_2 = 1011$ (เนื่องจาก $D_1 = 0100$ ไม่ว่าง)

ต่อมาสมมติว่า มีการยกเลิกการจัดสรร (Deallocation) ของบางหน่วยประมวลผลที่เสร็จจากงานที่ได้รับมอบหมาย คือ โหนดที่ 0000, 1001 ดังนั้น จะมี 8 โหนดชั้นนอกที่ว่าง (รูปที่ 8 ข)) และระบบต้องการหน่วยประมวลผลจำนวน $k=2^{m+2}=16$ หน่วยประมวลผล เพื่อรองรับงานที่จะเข้าสู่ระบบมาใหม่ แต่ไม่สามารถรวมหน่วยประมวลผลแบบ GCD [13] หรือ GCS [14] ได้ ดังนั้น เพื่อรองรับงานที่เข้ามาใหม่ จึงต้องใช้วิธีการรวมกลุ่มแบบเอ็กทรา-จีซีเอส ซึ่งสำหรับ 16 หน่วยประมวลผลที่มี $A = 0000$ จะได้ $D_1 = 0110$, $D_2 = 1001$ และ $D_3 = 1111$ ตามลำดับ ดังรูปที่ 8 ข)

5. ผลการวิจัย

ในงานวิจัยนี้มีการทดลองเพื่อประเมินประสิทธิภาพของการสื่อสารแลกเปลี่ยนข้อมูล โดยนำวิธีการสื่อสารเพื่อแลกเปลี่ยนข้อมูลแบบ ATAPE มาทดสอบความถูกต้องสำหรับการส่งข้อมูลจากหน่วยประมวลผลต้นทาง S สู่หน่วยประมวลผลปลายทาง D โดยการทดสอบจะใช้การเขียนโปรแกรมด้วยภาษาจาวา (JAVA Programming Language) และในการจำลอง HHCs จะมีข้อกำหนดดังต่อไปนี้ 1) การสื่อสารระหว่าง S และ D จะใช้เวลา 1 สัญญาณนาฬิกา (Clock) และ 2) การใช้เส้นเชื่อมบน HHCs ถ้าเป็นทิศทางตรงกันข้ามจะสามารถใช้เส้นเชื่อมเดียวกันได้

ตัวอย่างการทดลอง จะจำลองทิศทางการสื่อสารของโหนดต้นทาง S ไปยังโหนดปลายทาง D ซึ่งหน่วยประมวลผลทั้งหมดได้จากสมการที่ 2 ประกอบด้วย $A = 0000$ และ $D_1 = 0110$ ดังนั้นจะมีหน่วยประมวลผลทั้งหมด 8 หน่วยประมวลผล (โหนดละ 4 หน่วยประมวลผล) ประกอบด้วย 0000,00 (หรือ 0 ในเลขฐาน 10), 0000,01 (หรือ 1), 0000,10 (หรือ 2), 0000,11 (หรือ 3), 0110,00 (หรือ 24), 0110,01 (หรือ 25), 0110,10 (หรือ 26) และ 0110,11 (หรือ 27) ดังแสดงดังรูปที่ 9 สำหรับรอบการส่งข้อมูล หรือตัวควบคุมรอบจะถูกแทนด้วย C_i โดยจะมีทั้งหมด 8 รอบ (0 - 7) และการรับส่งข้อมูลจะเริ่มจากโหนดต้นทางที่จะถูกแทนด้วย $S = 0, 1, 2, 3, 24, 25, 26, 27$ และ โหนดปลายทางจะถูกกำหนดไว้ในตารางลาติน (Latin Square Table)



รูปที่ 9 ตัวอย่างตารางลาตินขนาด 8×8 เพื่อการสื่อสารแบบ ATAPE ของการแบ่งกลุ่มแบบเอ็กทรา-จีซีเอส

บน 6-HHCs ($N=64$, $m=2$) เมื่อ $A = 0000$, และ $D = 0110$

สำหรับการหาเส้นทางการสื่อสาร (Routing Paths) จะใช้ขั้นตอนวิธีที่ 1 ซึ่งถูกเสนอในงานวิจัย [13] และผลของเส้นทางการสื่อสารของตัวอย่างนี้ $A = 0000$ และ $D_1 = 0110$ ถูกแสดงในตารางที่ 3 โดยจะแสดงเฉพาะตัวควบคุมที่ 7 ($C = 7$) เพื่อการง่ายในการอธิบายและแสดงผล

ตารางที่ 3 แสดงเส้นทางการสื่อสารระหว่าง S และ D_1 ผ่าน 6 สัญญาณนาฬิกา (Clock) และ $C = 7$

Clock		1		2		3		4		5		6	
C=7	S = 0:0000,00	→	0000,01	→	0010,01	→	0010,00	→	0010,10	→	0110,10	→	27:0110,11 = D
	S = 1:0000,01	→	0010,01	→	0010,00	→	0010,10	→	26:0110,10 = D				
	S = 2:0000,10	→	0100,10	→	0100,11	→	0100,01	→	25:0110,01 = D				
	S = 3:0000,11	→	0000,10	→	0100,10	→	0100,11	→	0100,01	→	0110,01	→	24:0110,00 = D
	S = 24:0110,00	→	0110,01	→	0100,01	→	0100,00	→	0100,10	→	0000,10	→	3:0000,11 = D
	S = 25:0110,01	→	0100,01	→	0100,00	→	0100,10	→	2:0000,10 = D				
	S = 26:0110,10	→	0010,10	→	0010,11	→	0010,01	→	1:0000,01 = D				
	S = 27:0110,11	→	0110,10	→	0010,10	→	0010,11	→	0010,01	→	0000,01	→	0:0000,00 = D

ตารางที่ 3 แสดงให้เห็นว่าการรวมกลุ่มแบบเอ็กทรา-จีซีเอส สามารถจัดกลุ่มการสื่อสารแบบ ATAPE ระหว่างหน่วยประมวลผลได้โดยไม่มีการชนกันของข้อมูล สามารถสังเกตได้จากการไม่มีโหนดคั่นทาง และปลายทางซ้ำกันในช่วงของแต่ละสัญญาณนาฬิกา (Clock) นอกจากนั้นเส้นทางที่ได้ยังคงเป็นเส้นทางที่สั้นที่สุด ซึ่งจะมีประโยชน์อย่างมากในการสื่อสารแบบขนาน

สำหรับการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการแบ่งกลุ่ม จะทำการเปรียบเทียบดังตารางที่ 4 โดยจะเปรียบเทียบกับวิธีการแบ่งกลุ่มแบบ GCD การแบ่งกลุ่มแบบ GCS และการแบ่งกลุ่มแบบเอ็กทรา-จีซีเอส ซึ่งถูกเสนอในงานวิจัยนี้ ตารางที่ 4 แสดงให้เห็นว่างานวิจัยนี้มีประสิทธิภาพที่เพิ่มขึ้น โดยสามารถรองรับคุณสมบัติทั้งหมด โดยเฉพาะความสามารถในการใช้หน่วยประมวลผลภายใต้สถานการณ์ที่มีข้อจำกัด หรือสถานะที่หน่วยประมวลผลในระบบถูกใช้งานอยู่เป็นจำนวนมากได้

ตารางที่ 4 การเปรียบเทียบประสิทธิภาพของขั้นตอนวิธี

Features	GCD [13]	GCS [14]	Extra-GCS
Routing (Arbitrary S)	✓	✓	✓
Routing-conflict Solution	✓	✓	✓
Group Partitioning	✓	✓	✓
Maximum Task Sizes	$k \leq 2^{m+1}$	$2^{m+2} \leq k \leq 2^M$ เมื่อ $M = 2^m - 2^{m-1} + m - 1$	$2^{m+1} \leq k \leq 2^M$ เมื่อ $M = 2^m - 2^{m-1} + m - 1$
Parallel Applications	ATAPE	ATAPE	ATAPE
Critical Situations	-	-	✓

หมายเหตุ เครื่องหมาย ✓ หมายถึง สามารถรองรับคุณสมบัตินั้น ๆ ได้

6. สรุป

งานวิจัยฉบับนี้เสนอการรวมกลุ่มของหน่วยประมวลผลเพื่อการสื่อสารแบบขนาน บน n -HHCs ที่เรียกว่าการรวมกลุ่มแบบเอ็กทรา-จีซีเอส โดยการรวมกลุ่มนั้นถูกเสนอมาเพื่อรองรับในกรณีที่เครือข่ายมีหน่วยประมวลผลถูกใช้งานอยู่เป็นจำนวนมาก หรือภายใต้สถานการณ์ที่มีข้อจำกัดที่ไม่สามารถจัดสรรหน่วยประมวลผลสำหรับงานใหม่ที่จะเข้ามาสู่เครือข่ายได้ สำหรับความถูกต้องของการสื่อสารจะถูกตรวจสอบโดยใช้การสื่อสารแบบ ATAPE ซึ่งเป็นการสื่อสารแบบขนานที่มีประสิทธิภาพชนิดหนึ่ง ผลการทดลองพบว่า การรวมกลุ่มแบบเอ็กทรา-จีซีเอส สามารถรวมกลุ่มของหน่วยประมวลผลขนาด $2^{m+1} \leq k \leq 2^M$ เมื่อ $M = 2^m - 2^{m-1} + m - 1$ ได้อย่างมีประสิทธิภาพ และเมื่อมาทดสอบกับการสื่อสารแบบ ATAPE ก็สามารถส่งข้อมูลจากต้นทาง ถึงปลายทางได้อย่างถูกต้อง และยังคงมีคุณสมบัติของการเป็นเส้นทางที่สั้นที่สุดอยู่ ดังนั้นเมื่อเปรียบเทียบกับการจัดกลุ่มที่ถูกเสนอมาก่อนหน้านี้ คือ แบ่งกลุ่มของคิวอัลคิวแบบไขว้ (GCD) และการรวมกลุ่มคิวย่อยแบบไขว้ (GCS) พบว่า การรวมกลุ่มแบบ GCD และ GCS ยังไม่สามารถรองรับการใช้หน่วยประมวลผลภายใต้สถานการณ์ที่มีข้อจำกัดได้

7. ข้อเสนอแนะ

การเพิ่มประสิทธิภาพในการจัดสรรหน่วยประมวลผลบน HHCs จำเป็นที่จะต้องมีส่วนขั้นตอนวิธีในการจัดสรรหน่วยประมวลผลที่ทำการรวมหน่วยประมวลผลขนาดเล็กที่สุด เช่น 1 หรือน้อยกว่า 2^{m+1} หน่วยประมวลผล ซึ่งอาจอยู่คนละ Group กัน นำมารวมกลุ่มกันให้สามารถประมวลผลงานที่ต้องการได้ ซึ่งจะทำให้เกิดการใช้ประโยชน์จากหน่วยประมวลผลได้อย่างคุ้มค่าที่สุด

8. เอกสารอ้างอิง

- [1] ชัญญาวีจน์ สติฉิทรสมบัติ และ ภูมิพัฒน์ ดวงกลาง. (2564). การประยุกต์ใช้เทคโนโลยีภูมิสารสนเทศและการส่งข้อมูลผ่านเครือข่ายเซลลูลาร์. *วารสารวิทยาศาสตร์และเทคโนโลยีนายเรืออากาศ*, 17(1): 60-70.
- [2] Abd-El-Barr, M., & Al-Somani, T.F. (2011). Performance comparison of hierarchical interconnection networks. *IEEE Pacific Rim Conference on Communications*.: 191 – 196.
- [3] Antoine, B., & Keiichi, K. (2011). Set-to-Set Disjoint-path Routing in Perfect Hierarchical Hypercube. *The Fourth International C* Conference on Computer Science and Software Engineering*. : 51-57.
- [4] Cheng-Nan, L. (2014). An efficient construction of one-to-many node-disjoint paths in folded hypercubes. *J.Parallel and Distributed Computing*. vol.74: 2310-2316.
- [5] Cheng-Nan, L. (2017). Optimal Construction of Node-Disjoint Shortest Paths in Folded Hypercubes. *J. Parallel and Distributed Computing*. vol.102: 37-41.
- [6] Chienhua, C., & Dharma, P. A. (1993). dBCube: a new class of hierarchical multiprocessor interconnection networks with area efficient layout. *IEEE Trans. Parallel and Distributed System*. 4(12): 1332-1344.
- [7] Baojun, Q., Feng, S., & Weixing, Ji. (2007). THIN: A New Hierarchical Interconnection Network-on-Chip for SOC. *The 7th international conference on Algorithms and architectures for parallel processing*. : 446-457.
- [8] Ke, Q. (2008). An efficient Disjoint Shortest Paths Routing Algorithm for the Hypercube. *14th IEEE International Conference on Parallel and Distributed Systems*. : 43 – 47.
- [9] Yousef, S., & Martin, H.S. (1988). Topological properties of hypercubes. *IEEE Trans. Comput*. 37(7): 867-872.
- [10] Qutaibah, M., & Magdy, B. (1994). The hierarchical hypercube: A new interconnection topology for massively parallel systems. *IEEE Trans. Parallel and Distributed Systems*. 5(1): 17-30.
- [11] Ruei, Y.W., Chen, G.H., & Gen, J. (2007). Node-disjoint paths in hierarchical hypercube networks. *Information Sciences*. 117(19): 4200-4207.

- [12] Antoine, B., Keiichi, K., & Shietung, P. (2011). Node-to-set disjoint-path routing in perfect hierarchical hypercubes. *International Conference on Computational Science*. Vol.4: 442-451.
- [13] Nuntipat, P., & Jeeraporn, W. (2021). Shortest-Path Routing for Optimal All-to-All Personalized-Exchange Embedding on Hierarchical Hypercube. *J. Parallel Distrib. Comput.* Vol.150C: 139-154.
- [14] Nuntipat, P., & Jeeraporn, W. (2022). Optimal ATAPE-Task Scheduling on Reconfigurable and Partitionable Hierarchical Hypercube Networks. *Parallel Computing*. Vol.111: 102923.
- [15] Roselin, P., & Jeeraporn, W. (2017). Embedding the optimal all-to-all personalized exchange on multistage interconnection networks⁺. *J. Parallel Distrib. Comput.* 16(88): 1-15.
- [16] Scott, D.S. (1991). Efficient all-to-all communication patterns in hypercube and mesh topologies. *Distributed memory computing conference*. : 398-403.
- [17] Antoine, B., & Keiichi, K. (2013). k – pairwise disjoint paths routing in perfect hierarchical hypercubes. *The J. Supercomputing*. Vol 67: 485-495.