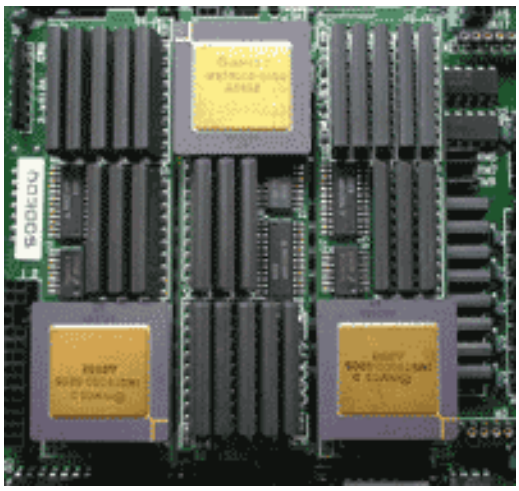
Transputer

น.อ.ผศ.ภาณุฤทธิ์ ยุกตะทัต
ผู้ช่วยศาสตราจารย์ ฝ่ายศึกษา โรงเรียนนายเรือ

คำว่า “Transputer” เป็นคำย่อของ “Transistor Computer” เป็น Microprocessor Chip ที่ใช้เทคโนโลยีในการสร้างเป็นแบบ Block เช่นเดียวกับการออกแบบทรานซิสเตอร์ในอดีต Transputer เป็นชิพที่มีพื้นฐานการออกแบบด้วยวงจรรวมขนาดใหญ่มาก (Very Large Scale Integrated-Circuit - VLSI) คุณสมบัติพิเศษของ Transputer ก็คือมีการเพิ่มระดับของการจัดการให้สามารถออกแบบระบบการคำนวณแบบขนาน (Parallel Processing) ได้ง่ายโดยต่อเป็นเครือข่าย Multi-Processor ดังนั้น Transputer จึงเป็น Single-Chip Microprocessor ที่มีหน่วยความจำท้องถิ่น (Local Memory) ภายในตัวเอง รวมทั้งมี Communication Links ในการติดต่อสื่อสารกับ Transputer ตัวอื่นโดยที่ไม่ต้องใช้เวลาการทำงานของหน่วยประมวลผลกลาง หรือซีพียู (CPU) นั่นคือ Transputer จะสามารถส่งข้อมูลได้ในขณะที่ซีพียูกำลังทำงานอยู่ นอกจากนั้นยังมีหน่วยคำนวณเลขทศนิยม (Floating Point Unit หรือ FPU) แยกทำงานอยู่ต่างหาก ทำให้สามารถคำนวณเลขทศนิยมได้ถึง ๖๔ หลักได้อย่างรวดเร็ว[๑] ดังนั้นถึงแม้ว่าในปัจจุบันชิพตระกูล Transputer จะนับว่าได้สูญพันธุ์ไปแล้ว แต่ด้วยคุณสมบัติอันทรงประสิทธิภาพเหล่านี้ Transputer โดยเฉพาะอย่างยิ่ง IMS T800 ซึ่งยังคงมีใช้งานอยู่ในระบบขนาดใหญ่หลายๆ ระบบจึงยังคงอยู่ในความทรงจำของเราตลอดไป



ในปี ค.ศ. ๑๙๘๕ (พ.ศ.๒๕๒๘) บริษัท INMOS ประเทศอังกฤษ ได้เปิดตัว Transputer (***Trans***istor Com***puter***) ซึ่งเป็นนวัตกรรมใหม่ในขณะนั้น เป็นครั้งแรกที่หน่วยประมวลผลได้ถูกนำมาประมวลผลรวมเข้ากับระบบการสื่อสารย่อย Transputer ติดต่อสื่อสารกับเครื่องอื่น ๆ ผ่านทาง Serial Links ที่มีความเร็วสูง Transputer ได้ถูกนำมาใช้ในงานจัดทำตารางการทำงานให้แก่งานที่ต้องประมวลผลหลายๆ คำสั่ง แม้ว่าตอนแรก Transputer นั้นได้ถูกออกแบบมาเพื่อใช้กับเครื่อง Super Computer ส่วนมากที่นิยมใช้ก็จะเป็นระบบฝังตัว บ้างก็จะใช้ในการ

---

[๑] Shem-Tov Levi and Ashok K. Agrawala, *Real Time System Design*, 1990.

ประมวลผลเกม (เป็นเกมที่ต้องใช้ความละเอียดสูงในด้านการประมวลผล ๓ มิติ และระบบเสียงสเตอริโอ) ซึ่งจะมี Transputer เป็นหัวใจหลักในการประมวลผล

Transputer ง่ายต่อการนำไปทำเป็นระบบ Multi-transputer เนื่องจากมี Communication Link เป็นอิสระ ซึ่งได้รับความนิยมเป็นอย่างมากในงานที่ต้องใช้การคิดคำนวณแบบขนานโดยวิธีการ Message-passing เช่น การควบคุมหุ่นยนต์ การประมวลผลภาพ ฐานข้อมูล ซึ่งสามารถที่จะคำนวณได้อย่างรวดเร็ว

หลักการทำงานของ Transputer

Transputer เป็น Parallel Microprocessor ของบริษัท INMOS [๒] ที่จัดอยู่ในประเภทของคอมพิวเตอร์แบบ MIMD (Multiple Instruction Multiple Data) ทำให้สามารถ Execute โปรแกรมหลายโปรแกรมกับข้อมูลหลายชุดได้ในเวลาเดียวกัน Transputer จึงทำงานได้โดยอิสระ หรือนับว่าเป็น Processor Element ที่ถูกเชื่อมต่อกันด้วย Link ของตนเองเป็นเครือข่าย Transputer

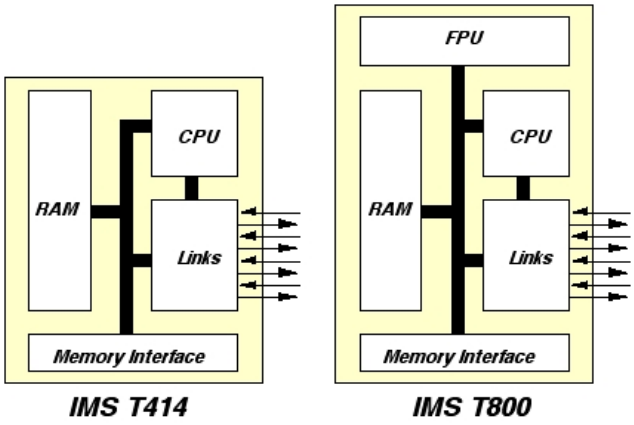
Transputer จะมีหน่วยประมวลที่ใช้สถาปัตยกรรมแบบ RISC (Reduced Instruction Set Computer) และก็มีการติดต่อกับระบบย่อยโดยใช้ ๔ Serial Links ที่มีความเร็วสูง มี RAM และมีตัวที่เชื่อมต่อกับหน่วยความจำอยู่บนชิพ ตารางข้างล่างแสดงให้เห็นคุณลักษณะของ Transputer ในตระกูลต่าง ๆ

คุณสมบัติ	T222	T225	M212	T414	T425	T800	T9000
สถาปัตยกรรม (bit)	16	16	16	32	32	32	64
รอบเวลาในการทำงานภายใน (ns)	50	50	50	50	33	33	20
การปฏิบัติคำสั่ง (MIPS)	20	20	20	20	30	30	100-200
FPU	No	No	No	No	No	Yes	Yes
SRAM on Chip (byte)	2K	2K	2K	2K	4K	4K	16K
Internal Bandwidth (M/วินาที)	80	80	80	80	120	120	200
Address space (byte)	64K	64K	64K	4G	4G	4G	4G
Interrupt Response (ns)	950	950	950	950	630	630	unstated
Link Speed (Mbit/วินาที)	20	20	20	20	20	20	100

[๒] ปัจจุบันควบรวมกิจการกับ บริษัท SGS-THOMSON Microelectronics

ถึงแม้ว่าการพัฒนา Transputer รุ่นสุดท้าย คือ T9000 ซึ่งเป็นไมโครโพรเซสเซอร์ที่มีประสิทธิภาพสูง แต่ในบทความนี้จะอธิบายหลักการทำงานของ Transputer T800 เนื่องจากเป็นไมโครโพรเซสเซอร์ซึ่งเป็นที่นิยมใช้อย่างแพร่หลายในสมัยหนึ่ง และสามารถทำความเข้าใจได้ไม่ยากนัก

INMOS IMS T800 Transputer ประกอบด้วยโพรเซสเซอร์ความเร็วสูง Floating-Point Processor หน่วยความจำภายใน และ Communication Links รวมอยู่ภายในชิพเพียงตัวเดียว โดยที่ IMS T800 ได้รับการออกแบบมาใช้ในการสร้างเครือข่าย Transputer โดยเฉพาะ จึงมีช่องการติดต่อสื่อสารข้อมูล (Communication Links) 4 Channels โดยที่แต่ละ Channel สามารถทำการติดต่อสื่อสารได้ ๒ ทิศทาง และมีการทำงานเหมือน DMA Channel[๓] (Direct Memory Access) จึงทำให้มีอัตราเร็วในการส่งข้อมูลสูงถึง 200 Mbps ต่อ Link โดย J. Dongarra ได้วัดความเร็วในการทำงานของ Transputer T800 25 MHz เปรียบเทียบกับ CPU ตัวอื่น ผลก็คือมีความเร็วในการทำงานมากกว่าซีพียู Pentium IV ของ Intel ประมาณ ๔ เท่า และมีความเร็วในการทำงานมากกว่าซีพียู M4 ของ Motorolla ประมาณ เท่า[๔]



รูปที่ ๑ Transputer T414 และ T800

Hardware Environment

ในการออกแบบเครือข่าย Transputer จะใช้เครื่องคอมพิวเตอร์ส่วนบุคคล (PC) ติดต่อกับ Host Transputer (T_0) IMS T800 25 MHz ที่อยู่บน TMB 08 Motherboard ทำหน้าที่ติดต่อระหว่างคอมพิวเตอร์ส่วนบุคคลกับเครือข่าย Transputer ที่ประกอบด้วย Transputer IMS T800 20 MHz อีก ๔ ตัวที่ต่อเป็นเครือข่ายบน IMS B009

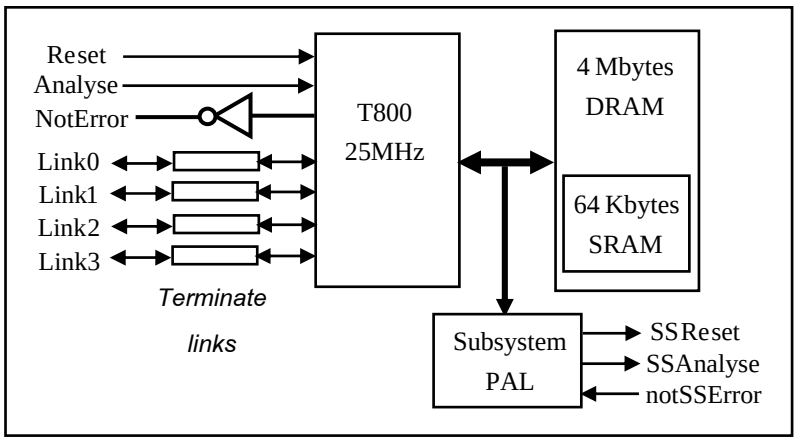
---

[๓] Thomas Richard McCalla, *Introduction to Numerical Methods*, 1967.

[๔] Uno R. Kodres, *Parallel Command and Decission System*, 1991.

๑. IMS B417 TRAM (TRAnsputer Module)

IMS B417 เป็น Transputer Module ซึ่งประกอบด้วย Transputer IMS T800 25 MHz หน่วยความจำแบบ Static RAM ขนาด 64 KB และ Dynamic RAM ขนาด 4 MB ใช้ในการ Run Alsis-Ada Compiler โดยเฉพาะ IMS B417 เป็น Transputer Board แบบง่ายที่ใช้เป็นมาตรฐานการเชื่อมต่อ สำหรับบล็อกไดอะแกรมของ IMS B417 แสดงตามรูปที่ ๒



รูปที่ ๒ IMS B417 TRAM

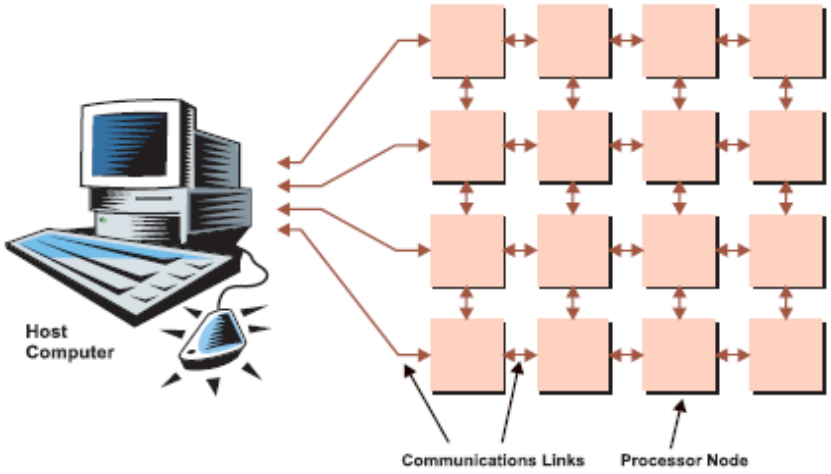
๑.๑ Transputer IMS T800 25 MHz เป็นไมโครคอมพิวเตอร์แบบ CMOS ขนาด ๓๒ บิต ที่มีหน่วยคำนวณเลขทศนิยม (Floating Point Unit) ขนาด ๖๔ บิต ใช้ในการคำนวณเลขทศนิยมแยกจากการทำงานของโพรเซสเซอร์ และสนับสนุนการทำงานใน Graphic Mode มีหน่วยความจำหลักภายใน (RAM On-Chip) ขนาด 4 K สำหรับการประมวลผลด้วยความเร็วสูง (แอดเดรส และดาต้า ขนาด ๓๒ บิต ทำให้มีอัตราเร็วในการขนส่งข้อมูลได้ 40 Mbps) IMS T800 มีการต่อเชื่อมกับหน่วยความจำของ IMS B417 ที่สามารถตั้งค่าการทำงานได้ รวมทั้งมี Link สำหรับการสื่อสารข้อมูล ที่ทำงานอย่างอิสระอีก 4 Links

๑.๒ โครงสร้างของหน่วยความจำ (Memory Configuration) IMS B417 สามารถที่จะ Access หน่วยความจำหลักได้ 4 MB ประกอบด้วย หน่วยความจำภายใน Transputer ขนาด 4 KB หน่วยความจำ SRAM ขนาด 60 KB และหน่วยความจำ DRAM ขนาด 1024 KB ซึ่งในความเป็นจริงแล้วมีการเพิ่มแอดเดรสของหน่วยความจำแต่ละประเภท จึงทำให้มีหน่วยความจำให้ใช้ได้ คือหน่วยความจำ SRAM ขนาด 64 KB และหน่วยความจำ DRAM ขนาด 4 MB ซึ่งเพียงพอสำหรับ Alsis-Ada Compiler[๕]

[๕] Compilation System สำหรับโปรแกรมภาษา ADA ของบริษัท Alsis

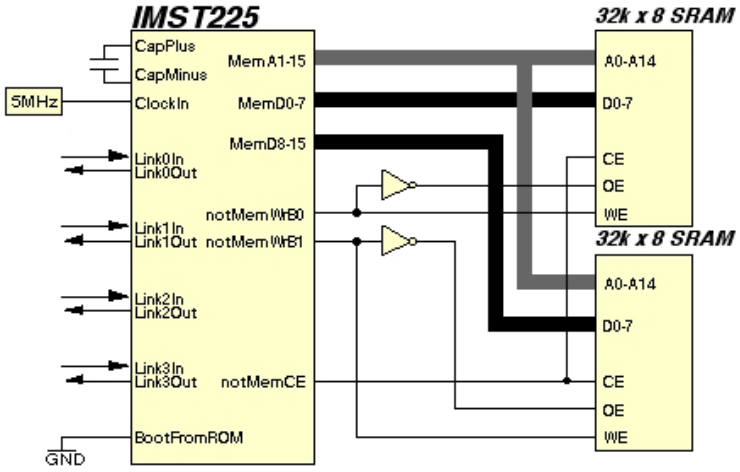
๒. TransTech TMB08 TRAM Motherboard

TMB08 คือ Motherboard ที่ทำหน้าที่เป็น Interface ระหว่างเครื่องคอมพิวเตอร์ส่วนบุคคลกับเครือข่าย Transputer โดยการส่งข้อมูลผ่านทาง PC Bus มี Slot สำหรับต่อ Transputer ได้สูงสุด ๑๐ ตัว TMB08 สามารถสร้างสัญญาณ Interrupt ไปยังเครื่องคอมพิวเตอร์ส่วนบุคคลเมื่อต้องการติดต่อกับผู้ใช้ และเมื่อยังไม่ต้องการติดต่อกับผู้ใช้ก็สามารถทำงานได้เองโดยอิสระและเครื่องคอมพิวเตอร์ส่วนบุคคลก็จะสามารถทำงานอื่นได้ และหากใช้ร่วมกับ IMS C104 programmable link switch ในการต่อเชื่อมจะทำให้สามารถสร้างเครือข่าย Transputer ขนาดใหญ่ภายใต้การควบคุมของโปรแกรมได้ การทำงานของ IMS C104 จะถูกควบคุมโดย IMS T225 Transputer ซึ่งอยู่บน TMB08



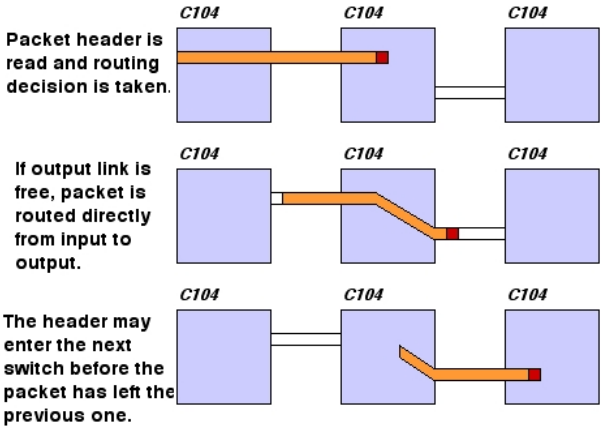
รูปที่ ๓ Transputer Network

๒.๑ IMS T225 Transputer เป็นไมโครโพรเซสเซอร์ขนาด ๑๖ บิต ประกอบด้วย Link มาตรฐานในการติดต่อสื่อสาร 4 Links หน่วยความจำหลัก 64 KB ซึ่งควบคุมการทำงานด้านการติดต่อสื่อสารทั้งหมดของเครือข่าย Transputer ที่ต่อเชื่อมอยู่กับ TMB08 Motherboard



รูปที่ ๔ IMS T225

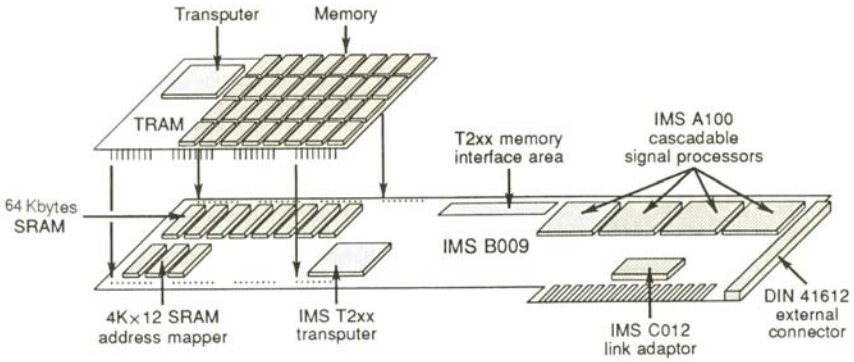
๒.๒ IMS C104 Programmable Link Switch เป็น Crossbar Switch ที่ได้รับการออกแบบมาให้สามารถส่งข้อมูลขนาด ๓๒ บิต ระหว่าง Transputer ที่ต่อเป็นเครือข่ายบน TMB08 Motherboard



รูปที่ ๕ การกำหนดเส้นทางการขนส่งข้อมูลของ IMS C104

๓. IMS B009 Evaluation Board

IMS B009 ประกอบด้วย Transputer T800 20 MHz จำนวน ๔ ตัวต่อเชื่อมกันเป็นเครือข่าย และถูกทำให้ Synchronize กันโดยใช้ System Clock เดียวกัน มี IMS C012 Link Adapter เป็นตัวควบคุมการเชื่อมต่อ Transputer กับ Host Computer

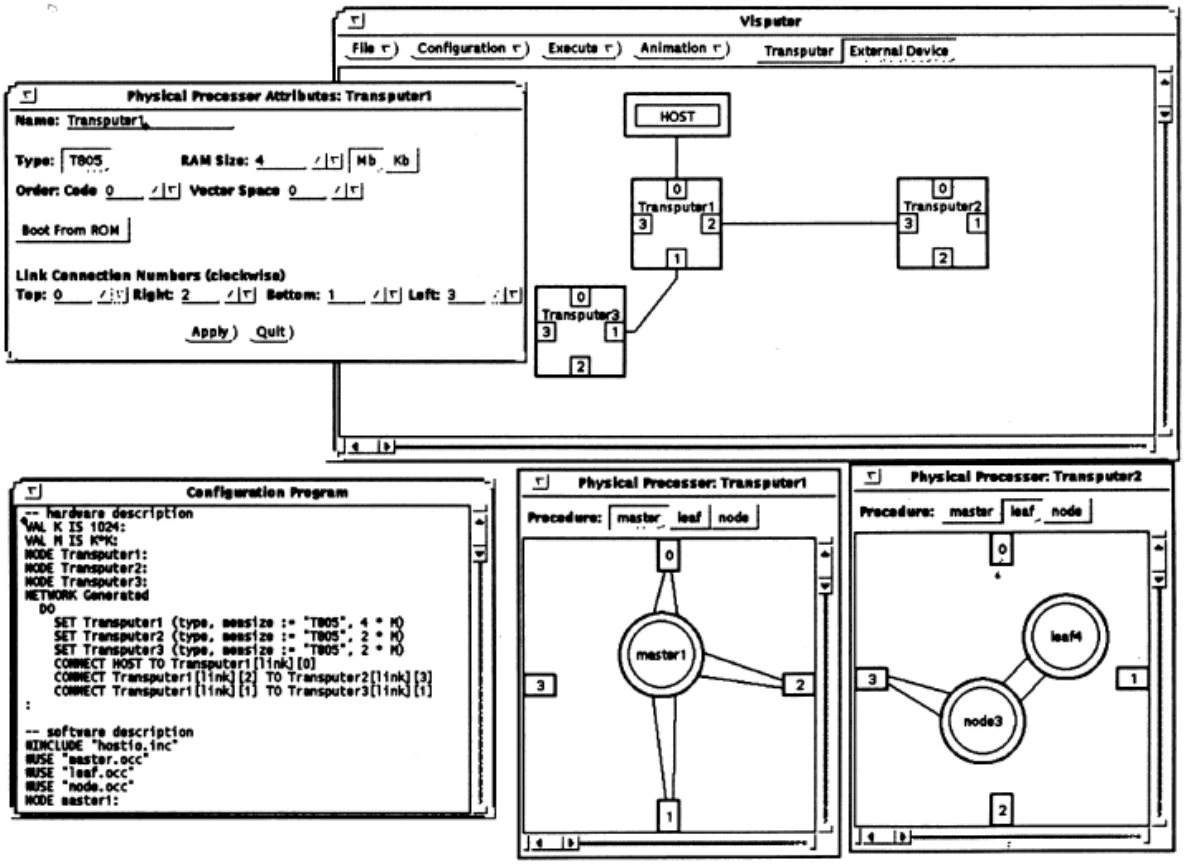


รูปที่ ๖ IMS B009 Evaluation Board

๓.๑ IMS T800 20 MHz Transputer เป็นไมโครโพรเซสเซอร์ ขนาด ๓๒ บิต ซึ่งประกอบด้วย Floating Point Unit (FPU) ขนาด ๖๔ บิต ที่มี Micro Code ควบคุมการทำงานอย่างอิสระให้สามารถทำการคำนวณได้ตลอดเวลาที่ได้รับข้อมูล มี Communication Link 4 Links ไว้สำหรับเชื่อมต่อ Transputer เข้าด้วยกันหรือติดต่อกับภายนอกโดยอิสระซึ่งสามารถส่งข้อมูลระหว่างกันด้วยความเร็วสูง และหน่วยความจำขนาด 4 KB ซึ่งใช้วิธีการ Multiplex ทั้ง Data Line และ Address Line ทำให้สามารถสนับสนุนการทำงานอย่างอิสระของ FPU และ Communication Link ได้พอเพียง

๓.๒ IMS C012 Link Adapter มีหน้าที่บริหารจัดการการติดต่อสื่อสารระหว่าง TMB08 Motherboard กับเครื่องคอมพิวเตอร์ส่วนบุคคล ให้เป็นแบบ Full Duplex

๓.๓ มี Transputer ๔ ตัว ต่อกันเป็นเครือข่ายแบบ Square ผ่านทาง Linkโดยที่ Link2 ของ Transputer ตัวหนึ่งจะต่อกับ Link3 ของ Transputer ตัวถัดไป ส่วน Link0 และ Link1 ของ Transputer แต่ละตัวใช้เป็น Input/Output ความเร็วในการส่งข้อมูลระหว่าง Link ต่าง ๆ ภายใน IMS B009 อยู่ที่ 20 Mbps ในขณะที่เมื่อต่อ Link0 ไปยังภายนอก จะส่งข้อมูลด้วยความเร็ว 10 Mbps อย่างไรก็ตาม เราสามารถตั้งให้ความเร็วในการส่งข้อมูลเป็น 10 Mbps หรือ 20 Mbps ก็ได้ ส่วนการควบคุมการทำงานของ Link ที่ต่อไปภายนอกสามารถทำได้โดยใช้โปรแกรมภาษา Occam ช่วยในการควบคุมผ่านโปรแกรมประยุกต์แบบกราฟิก ชื่อ Visputer ดังรูปที่ ๗



รูปที่ ๗ A Configuration created using Visputer

Software Environment

กระทรวงกลาโหมสหรัฐอเมริกาต้องสูญเสียงบประมาณเป็นจำนวนมหาศาล ในการจัดหาซอฟต์แวร์มาใช้กับระบบคอมพิวเตอร์ต่าง ๆ ในทางทหาร เนื่องจากไม่มีซอฟต์แวร์ในท้องตลาดที่ตรงกับความต้องการในด้านทางทหาร และยากต่อการพัฒนาต่อไปเนื่องจากไม่สามารถนำโมดูลเดิมมาใช้ในการเขียนโปรแกรม ใหม่ได้ ดังนั้นในกลางทศวรรษที่ ๗๐ จึงได้ประกาศความต้องการเบื้องต้นของซอฟต์แวร์ที่กระทรวงกลาโหมสหรัฐอเมริกาต้องการ คือ เป็นภาษาชั้นสูง ที่สามารถสนับสนุนงานพื้นฐานของวิศวกรรมซอฟต์แวร์สมัยใหม่ได้ เช่น Modularity, Strong Typing, Data Abstraction และ Information Hiding จึงได้มีการแข่งขันการพัฒนาภาษาคอมพิวเตอร์

โดยรวมความสามารถของภาษา ALGOL68, Pascal และ PL/1 เข้าด้วยกัน ในที่สุดในปี ค.ศ.๑๙๘๓ ผู้ที่ชนะการประกวด มีชื่อว่า Augasta Ada Loveless เป็นลูกสาวของ Lord Byron นักคณิตศาสตร์ชาวอังกฤษ และได้ประกาศให้เป็นภาษาคอมพิวเตอร์มาตรฐาน ANSI/MIL-STD-1815A ของกระทรวงกลาโหมสหรัฐอเมริกา (Department of Defence - DoD) โดยตั้งชื่อให้เป็นเกียรติแก่ผู้ชนะการประกวดว่า “ภาษา ADA[๖]” และในเดือนกุมภาพันธ์ ๑๙๙๕ โปรแกรมภาษา ADA 95 ได้การรับรองมาตรฐานให้เป็นภาษาที่ใช้สำหรับการเขียนโปรแกรมเชิงวัตถุ (Object-oriented Programming) เป็นภาษาแรกในโลก

๑. ADA Programming Language

๑.๑ ภาษา ADA Version 9.X[๗] เป็นภาษาคอมพิวเตอร์มาตรฐานของกระทรวงกลาโหมสหรัฐอเมริกา ที่ใช้ในการพัฒนาระบบอาวุธ เนื่องจากเป็นภาษาที่มีคุณสมบัติดังนี้

- เป็นภาษาแบบโครงสร้าง (Control Structure) ทำให้ง่ายต่อการพัฒนา
- สนับสนุนการทำงานแบบ Multi-tasking
- Strong Typing ตัวแปรที่ได้รับการประกาศให้เป็นประเภทใดประเภทหนึ่งแล้ว จะมีลักษณะเฉพาะที่ไม่สามารถนำไปใช้ร่วมกับตัวแปรประเภทอื่นได้
- Case Sensitive ตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กจะเป็นตัวอักษรคนละตัวกัน
- Information Hiding การซ่อนข้อมูลที่สำคัญเพื่อป้องกันการเข้ามาแก้ไข
- Exception Handling ช่วยแก้ไขข้อผิดพลาดขณะ Runtime
- Data Abstraction

๑.๒ การใช้ภาษา ADA

๑.๒.๑ ในประเทศสหรัฐอเมริกา

- ในระบบ Aegis Combat System
- โครงการ ATF (Advanced Tactical Fighter) Program
- NASA (National Aeronautical and Space Administration)
- FAA (Federal Aviation Associate)
- บริษัท Boeing ใช้ใน 7x7 Program
- CCA (Computer Corporate of America)

---

[๖] Jan Skansholm, *ADA From the beginning*, 1989.

[๗] Grady Booch, *Software Engineering with Ada*, 1988.

๑.๒.๒ ในยุโรป

- British Bank ใช้ ADA ด้วยเหตุผลในการรักษาความปลอดภัย
- Finnish Bank ใช้ ADA ด้วยเหตุผลในการรักษาความปลอดภัย
- รัฐบาลสเปน ใช้ ADA ในระบบการควบคุมการจราจรทางอากาศ
- NATO (North Atlantic Treaty Organization) ใช้ ADA ในระบบ C^3I (Command Control Communication and Information)

๒. Alsis-Ada[๏] Compilation System

บริษัท Alsis ได้ผลิต Compilation System ที่มีความสามารถในการสนับสนุน โปรแกรมการทำงานแบบขนานด้วยภาษา ADA เป็นครั้งแรกในปี ค.ศ.๑๙๘๙ ซึ่งประกอบด้วย Compiler และ Binder โดยที่ Compiler จะทำการสร้าง Executable Code จากโปรแกรมภาษา ADA สำหรับสั่งให้ Transputer ในตระกูล T4 และ T8 ทำงาน และมีความอ่อนตัวในการเรียกใช้ Libraries ต่างๆ ของภาษา ADA โดยใช้วิธีการแบ่งปันการใช้ร่วมกัน ไม่ต้องทำสำเนามาไว้ในโปรแกรม จึงทำให้ขนาดของโปรแกรมไม่ใหญ่มากจนเกินไป

๒.๑ Compiler

Compiler ของ Alsis-Ada ต้องการ Source File ๒ ไฟล์ เป็น Input คือ ตัวโปรแกรมภาษา ADA และโปรแกรม Library ที่จะสามารถเก็บออบเจ็กต์ที่คอมไพล์แล้วได้ ดังนั้นใน Source File จะต้องกำหนดลำดับของการทำงานว่า Unit ใดต้องทำงานก่อนและ Unit ใดจะทำงานทีหลัง โดยใช้คำว่า With เป็นตัวกำหนด หากไม่มีการกำหนดลำดับที่แน่นอน Compiler จะแสดงข้อความการผิดพลาดและจะหยุดการทำงาน โดยทั่วไป Output ของ Compiler จะเป็นการปรับปรุง Library ให้มีความทันสมัย ซึ่งประกอบด้วย Object Code ของแต่ละโปรแกรมใน Source file นอกจากนั้น Compiler ยังสามารถแสดงรายการความผิดพลาดที่เป็น Syntactic Error ในการคอมไพล์โปรแกรม เพื่อที่จะได้สามารถแก้ไขโปรแกรมได้ง่าย ส่วนความผิดพลาดที่เป็น Semantic Error นั้นผู้เขียนโปรแกรมจะต้องวิเคราะห์เอง

๒.๒ Binder

Ada Binder จะทำหน้าที่รวมออบเจ็กต์ต่างๆ ที่คอมไพล์แล้วซึ่งโปรแกรมต้องใช้ในการทำงาน ให้เป็นออบเจ็กต์โมดูลเดียวกัน Input ของ Binder ประกอบด้วย Ada program Library ที่บรรจุ Main Unit ของโปรแกรมภาษา ADA และชื่อของ Main Program โดย Binder จะใช้ชื่อของ Main Program ในการค้นหาออบเจ็กต์ต่างๆ ที่คอมไพล์แล้วใน Library รวมทั้ง Unit ที่มีส่วนเกี่ยวข้องกับ

[๏] Grady Booch, Op.cit.

โปรแกรมที่อยู่ใน Library อื่น แล้วนำมาดำเนินกระบวนการสร้าง Output ออกมา ๒ ไฟล์ คือ “program.o” เป็นออบเจ็กต์โมดูลที่รวบรวมข้อมูลและโปรแกรมภาษา ADA ซึ่งคอมไพล์แล้วทั้งหมด และ “program.bnd” ซึ่งเป็นรายการสรุปผลการดำเนินการตามกระบวนการ Binding ทั้งหมดที่ประกอบด้วยข้อความแสดงความผิดพลาด ข้อมูลเตือนการใช้ค่าตัวแปรต่างๆที่อาจก่อให้เกิดปัญหาในขณะ Run Time ได้ หากตรวจพบข้อผิดพลาดต่างๆ ในขั้นตอนนี้ก็จะไม่มีการสร้างออบเจ็กต์โมดูลอีก

๓. OCCAM 2 Toolset

เป็นเครื่องมือทางด้านซอฟต์แวร์ซึ่งใช้ในการพัฒนาโปรแกรมการทำงานของ Transputer ที่ทำหน้าที่เป็น Host ติดต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคล โดยจะใช้ Occam Libraries ในการจัดเตรียมสภาวะแวดล้อมของระบบเครือข่าย สำหรับการพัฒนาโปรแกรมที่ทำงานบน Transputer และเครือข่าย Transputer OCCAM 2 Toolset เป็นเครื่องมือที่อนุญาตให้เขียนโปรแกรมด้วย Text Editor ใดๆ ก็ได้ การ Run โปรแกรมประยุกต์ภาษา ADA จะต้องใช้โปรแกรมภาษา Occam ที่ฝังตัวอยู่บน Host Transputer

๓.๑ Toolset

ในขั้นตอนการ Linking และ Loading จะดำเนินการโดยใช้ Occam 2 toolset ที่ได้รับการพัฒนาโดยบริษัท INMOS มีเครื่องมือหลายอย่างให้เลือกใช้ตามประเภทของ Host Computer อย่างไรก็ตามเครื่องมือพื้นฐานที่จะใช้ มีดังต่อไปนี้

- ilink : ทำหน้าที่รวบรวมโปรแกรมทั้งหมดที่ใช้ในการวิเคราะห์
- iboot : เครื่องมือเพิ่ม Bootstrap เข้าไปในโปรแกรมก่อนที่จะ Load โปรแกรมเข้าไปใน Transputer
- iconf : เครื่องมือในการตั้งค่าให้โปรแกรมรู้จักเครือข่าย Transputer
- iserver : ทำหน้าที่เป็น Host Server ที่จะ Load Bootable Program เข้าไปยัง Processor

ในการใช้ Toolset เหล่านี้ต้องมีการกำหนดชื่อและนามสกุลของไฟล์ให้เป็นไปตามกฎที่กำหนดไว้ของโปรแกรมภาษา Occam คือ ชื่อไฟล์จะถูกกำหนดให้เป็นชื่อเดียวกับโปรแกรมภาษา ADA ส่วนนามสกุลส่วนใหญ่จะประกอบด้วยตัวอักษร 3 ตัว โดยอักษรตัวแรกจะถูกกำหนดดังนี้

- .cxx เป็นไฟล์ที่ได้จากการใช้ ilink (Link Code File)
- .mxx เป็น Mapping ที่ได้จากการใช้ ilink (Module Map)
- .bxx เป็น Bootable Code File ที่ได้จากการใช้ iboot
- .dxx เป็นคำอธิบาย Bootstrap ที่ได้จากการใช้ iboot

ส่วนตัวอักษรอีก ๒ ตัว (xx) นั้นอักษรแรกจะถูกกำหนดตาม Mode ความผิดพลาดในการ Compile โดยใน Alsis-Ada Compilation System จะใช้ STOP (s) Mode [๙] ในการหยุดการทำงานของกระบวนการ ดังนั้นตัวอักษรตัวที่ ๒ จึงเป็น “s” เสมอ สำหรับตัวอักษรตัวสุดท้ายจะถูกกำหนดตามประเภทของ Transputer ที่ใช้ โดยอาจจะเป็น “4” “5” หรือ “8” ขึ้นอยู่กับใช้ Transputer T414, T425 หรือ T800 ตามลำดับ

๓.๒ *Program Linking*

ในระบบเครือข่าย Transputer ที่ Run โปรแกรมภาษา ADA หลายโปรแกรม จำเป็นต้องใช้ ilink ในการเชื่อมโยงความสัมพันธ์ของโปรแกรมต่างๆ เข้าด้วยกัน โดยใช้ Syntax คือ

Ilink {inputfilr} {option}

และจะต้องมี input file ดังนี้

- harness.t8s : ไฟล์โปรแกรมภาษา Occam ที่ใช้ในการเรียกโปรแกรม ADA
- adarts8.lib : Library ที่ใช้ในการจัดลำดับการทำงานในขณะ Run Time
- occam8s.lib : Occam Compiler Library ที่จัดเตรียม Toolset ให้โปรแกรมเรียกใช้ตามลำดับที่กำหนดโดย adarts8.lib
- hostio.lib : Library ของ Occam 2 ที่จัดเตรียมการ aeecss ไปยัง Server

๓.๓ *Program Loading and Execution*

ถึงแม้ว่า Linker จะรวบรวมทุกสิ่งทุกอย่างเข้าไว้ในไฟล์เดียวกัน แต่ในการทำงานของเครือข่าย Transputer มีความจำเป็นต้องเตรียมการกระจายโปรแกรมต่าง ๆ ไปยัง Transputer แต่ละตัว โดยสร้าง Object นั้น ให้เป็น Executable Code ด้วยการใช้ iconf

๔. Make Program Maintenance Utility

Make เป็นโปรแกรมยูทิลิตี้ (Utility) ที่ได้รับการออกแบบมาสำหรับใช้ในการช่วยเหลือกระบวนการปรับปรุงโปรแกรมให้ทันสมัยอยู่ตลอดเวลาโดยอัตโนมัติ จึงจัดเตรียมให้มีภาษาสคริปต์ (Script) สั้นๆ ที่ใช้ Interpreter โดยเขียนไว้ในไฟล์ชื่อว่า “makefile” และจะเริ่ม Execute รายการข้อมูลที่มีใน Makefile ตามลำดับ เพื่อตรวจว่ามีส่วนใดของโปรแกรมที่ได้รับการปรับปรุงหลังจากที่คอมไพล์แล้วและมีผลกระทบต่อโปรแกรมอื่น เมื่อตรวจพบก็จะปรับปรุงให้ Main Program ทันสมัยอยู่ตลอดเวลา

การเรียกใช้ Make สามารถทำได้โดยการใช้คำสั่ง #IMPORT

---

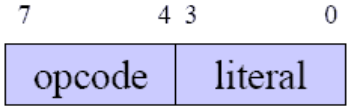
[๙] INMOS Limited, *Transputer Development and **iq** Systems Databook,* 1989.

หลักการนำเทคโนโลยีการคำนวณแบบขนานมาใช้ในการออกแบบโปรแกรม

ในการออกแบบระบบต่างๆ โดยนำเทคโนโลยีการคำนวณแบบขนานมาใช้ สิ่งที่จำเป็นคือการออกแบบให้มีการแบ่งปันการใช้ทรัพยากรต่าง ๆ ร่วมกัน จึงจำเป็นต้องควบคุมการติดต่อสื่อสารทั้งภายใน Transputer และการติดต่อสื่อสารระหว่างกันผ่านทาง Transputer Port

๑. Transputer's Instruction Set

Transputer ทุกตัวมีพื้นฐานของ Instruction Set เหมือนกัน ซึ่งจะมีหมายเลขของแต่ละ Instructions ที่ใช้เป็นตัวแทนในการทำงานบ่อยที่ปรากฏในโปรแกรมแต่ละ Instruction ประกอบไปด้วยข้อมูลยาว ๑ ไบต์ (๘ บิต) ซึ่งเป็นแบบ Short Fixed Length โดย แบ่งออกเป็น ๔ บิต ที่ เป็น 'nibble' ซึ่งเป็นชื่อของคำสั่งโดยคำสั่งที่ใช้บ่อยเป็นพื้นฐานมี ๑๖ ตัว เช่น loads, stores, jumps, calls เป็นต้น



ทุกคำสั่งจะมีความยาวอย่างน้อย ๔ บิต เป็นของ Operand Register แต่ละคำสั่งจะใช้ ๑ Operand โดย Operand จะอยู่ใน Register และ Operand Register ถูกใช้เป็น Operand ของคำสั่งที่มีลักษณะพิเศษของ Function Code คำสั่งทั่วไปจะล้างค่าของ Operand Register หลังจากการดำเนินการ อย่างไรก็ตาม Prefix Instructions ไม่ได้ไปทำอะไรที่เกี่ยวข้องกับ ๔ บิต ใน Operand Register และ Negative Prefix Instructions เป็นส่วนเสริมในการบรรจุของ Operand Register หลังการการเลื่อนในการดำเนินคำสั่งจะใช้ตัว Operand ราวกับเป็น Opcode เพื่อให้ Transputer เข้าถึงคำสั่งพื้นฐานทั้ง ๑๖ คำสั่ง เมื่อ Operand ได้มีการเติมเข้าไปด้านหน้า ทำให้ Transputer มีหมายเลข Instructions ได้ตามใจชอบ แต่ถูกจำกัดเพียง ๑๓ Instructions ที่สามารถมี Immediate Operand นอกนั้นต้องมี Operand อยู่ด้วย เช่น คำสั่ง Add ประกอบด้วย A และ B โดยทำการบวกกันและเก็บค่าผลลัพธ์ไว้ที่ A

IMS T800 มีการเพิ่ม Instructions เข้าไป โดยมีคำสั่งที่สามารถเก็บค่าจาก Floating-Point Register Stack ซึ่งเป็นตัวที่สนับสนุนเกี่ยวกับเรื่องกราฟิก ซึ่งมีรูปแบบที่ยอมรับกัน และอีกจุดประสงค์คือเป็นเครื่องมือในการทำ Error Correction

ตัว Operand ใน Transputer จะถูกใช้กระทำโดยตรง (Immediate Addressing) ไม่มีการอ้างอิงถึงหน่วยความจำนอกเหนือจากการอ่านเข้ามาในซีพียู

สำหรับการติดต่อสื่อสารทั้งภายในและระหว่าง Transputer เนื่องจากโปรแกรมภาษา ADA ยังไม่สนับสนุนการแบ่งปันการใช้ทรัพยากรร่วมกันภายใน Transputer ดังนั้นการที่จะทำให้ฟังก์ชันการทำงานที่ต้องการ ให้ทำงานพร้อมกันในเวลาเดียวกัน จะต้องอาศัยโปรแกรมภาษา Occam มาสนับสนุน ซึ่ง

โครงสร้างโปรแกรมภาษา Occam (Harness) จะประกอบด้วยการส่งกระบวนการต่าง ๆ ที่มีอยู่เป็นจำนวนมากไปยังเนื้อที่ว่างของหน่วยความจำหลัก ๒ พื้นที่ คือ Work Space และ Stack Memory จึงเป็นหน้าที่ของผู้เขียนโปรแกรมที่จะทำการแบ่งการทำงานต่าง ๆ ไปยังหน่วยความจำหลักที่เป็น Work Space ตามความเหมาะสม

๒. การควบคุมการติดต่อสื่อสารภายใน Transputer

ในการควบคุมการติดต่อสื่อสารภายใน Transputer มีขั้นตอนการดำเนินการ ๔ ขั้นตอนดังนี้

๒.๑ Source Compilation

โปรแกรมทั้งหมดจะต้องได้รับการคอมไพล์ก่อนที่จะทำการ Link ได้ การเลือกใช้ Alsis-Ada Compilation System จะทำให้สามารถแยกภาษา ADA ออกจากโปรแกรมภาษา Occam และสร้าง Object Code ที่สามารถทำการ link ได้ ในการพัฒนาจึงมี Source File คือ

- Source of Occam Harness : จะเป็นตัวกำหนด Channel การติดต่อสื่อสารของโปรแกรมภาษา ADA โดยที่ในโปรแกรมภาษา ADA จะมีส่วนที่เรียกว่า Mini Harness ให้ Main harness เรียกใช้ให้ทำงานพร้อมกัน โดยไฟล์นี้จะต้องมีนามสกุล ".occ"
- Source of ADA Program : เป็นโปรแกรมประยุกต์ที่เขียนด้วยภาษา ADA ที่จะ Run บน Transputer การที่จะออกแบบให้ Transputer นั้นๆ ทำงานเป็นระบบใดก็จะต้องเขียนโปรแกรมประยุกต์ตามฟังก์ชันการทำงานนั้นๆ โดยไฟล์นี้จะต้องมีนามสกุล ".ada"

๒.๒ Object Linking

ถึงแม้ว่าการเรียกใช้เครื่องมือ Binder ก็เพียงพอในการ Run โปรแกรมภาษา ADA โปรแกรมเดียว บน Transputer ตัวเดียว และสามารถสร้างไฟล์ "proj.o" ได้ แต่อย่างไรก็ตามก็ยังจำเป็นต้องมีการเรียกใช้ข้อมูลบางส่วนใน Library ในขณะ Run-time จึงต้องใช้เครื่องมือ Ilink ในการเชื่อม Library เหล่านั้น และรวมเข้าเป็นออบเจ็กต์โมดูลเดียวกัน ปัญหาที่อาจเกิดขึ้นได้ในการ Link โปรแกรมเพื่อที่จะ Run บน Transputer ตัวเดียวคือการใช้ชื่อซ้ำกัน ซึ่งสามารถหลีกเลี่ยงได้โดยในการเขียนโปรแกรมภาษา ADA ให้กำหนดจุดที่จะทำการ Link โดยใช้ตัวเลือกในเครื่องมือ Binder คือ คำว่า ENTRY_POINT

๒.๓ Configuring

ถึงแม้ว่า Linker จะสร้างออบเจ็กต์โมดูลที่พร้อมจะ Run บน Transputer แล้ว แต่ก็ยังมีอีกขั้นตอนหนึ่งก่อนที่จะ Run โปรแกรม คือการ Configure ให้โปรแกรมเตรียมที่จะ Run บน Transputer เป้าหมาย โดยการเพิ่ม Bootstrap Code เข้าไปในโปรแกรม เพื่อที่จะกำหนดค่าเริ่มต้นของ Transputer ตัวที่จะ Run โปรแกรม และกำหนดตำแหน่งของหน่วยความจำที่จะใช้เป็น Work Space การที่โปรแกรมจะเรียกใช้ Bootstrap Loader จะใช้ Syntax ดังนี้

```
PROC main.program (CHAN OF ANY from.server, to.server,
                   [  ]INT workspace, stack.memory)
```

ตัวแปร from.server และ to.server คือ Channel ที่ Host File Server ใช้ในการ Boot โปรแกรม และอาจจะถูกใช้ในการติดต่อสื่อสารกับ Server ในขณะ Run-time ด้วย สำหรับตัวแปร Workspace และ Stack.Memory เป็นพื้นที่หน่วยความจำที่ว่าง ที่ถูกกำหนดให้โปรแกรมประยุกต์ใช้งาน

๒.๔ *Loading and Executing*

ไฟล์ที่ถูกสร้างโดยเครื่องมือ Iboot จะเป็น Executable File ที่จะใช้ Run บน Transputer และจะทำการ Load โปรแกรมไปยัง Transputer เป้าหมายโดย Host File Server ด้วยการใช้เครื่องมือ Iserver

ดังนั้นคำสั่งดังต่อไปนี้ จึงต้องเตรียมให้มีอยู่ใน Makefile เพื่อใช้ในการสร้างสภาวะแวดล้อมของระบบและ Run โปรแกรมภาษา ADA โดยสมมติว่าโปรแกรมภาษา ADA นั้นได้รับการคอมไลพ์แล้ว และถูกสร้างให้อยู่ในรูปของไฟล์ proj.o (รูปที่ ๘) ซึ่งมีขั้นตอนในการทำงานดังนี้

- ทำการ compile โปรแกรมภาษา Occam ที่ใช้สนับสนุนการทำงาน

```
occam /s /t8 merger.occ
occam /s /t8 projh.occ
occam /s /t8 projh2.occ
occam /s /t8 main.occ
```

- ทำการ Link

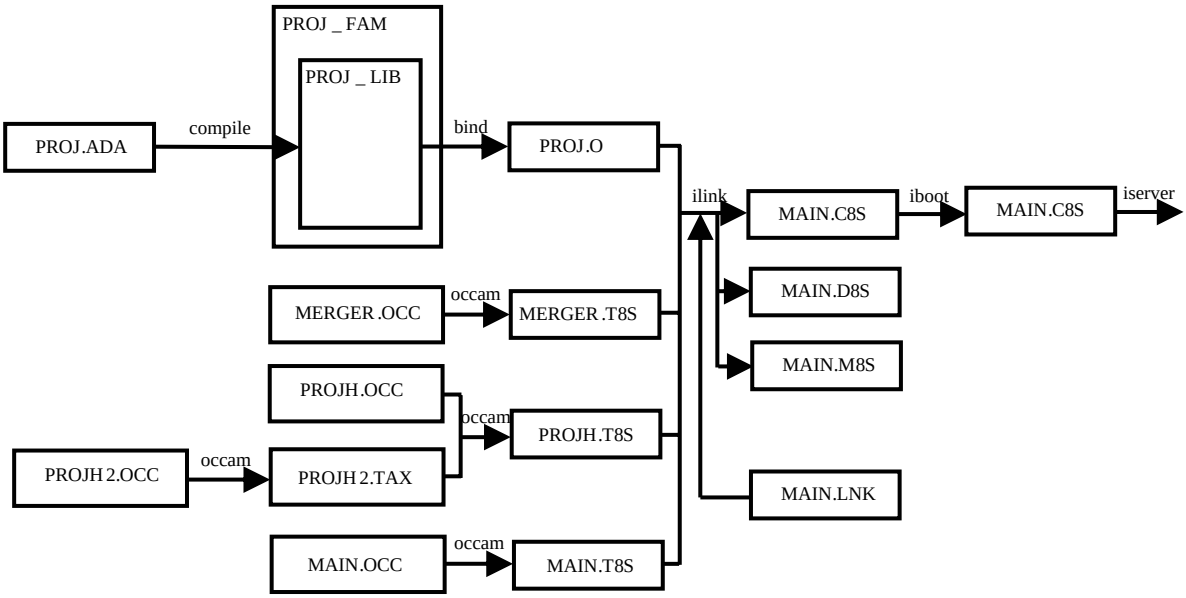
```
ilink /f main.lnk
```

- ทำการเพิ่ม Bootstrap

```
iboot main.c8s
```

- ทำการ Load และ Run โปรแกรม

```
iserver /sb main.b8s
```



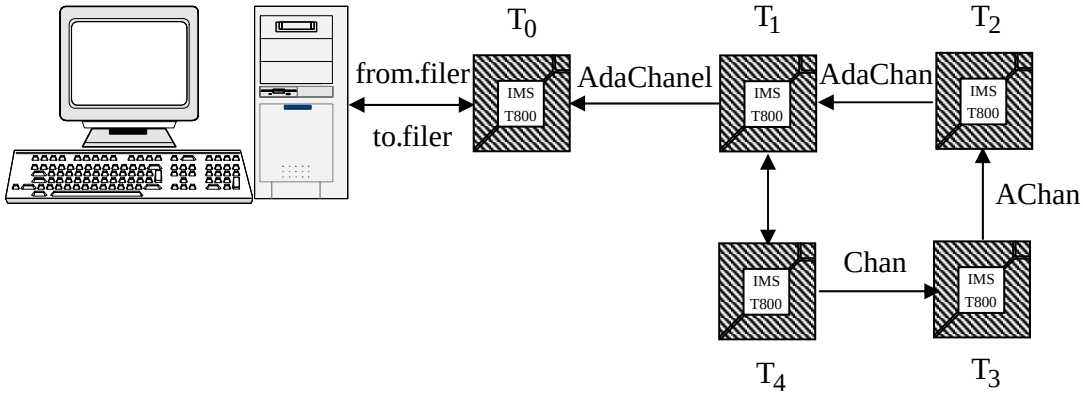
รูปที่ ๘ ขั้นตอนการทำงานของ Transputer

๓. Transputer Port

ในสถาปัตยกรรม I/O Port ที่เด่นชัดที่สุดคือ Memory Mapped หมายถึงว่าการโปรแกรมให้ใช้งานพอร์ตหนึ่ง ๆ รวมถึงการส่งแอดเดรสและความยาวของข้อมูลจะทำได้โดยการเขียนเลขนี้เพื่อเก็บตำแหน่งของหน่วยความจำเอาไว้ Mapped ไปที่ Register ของพอร์ต

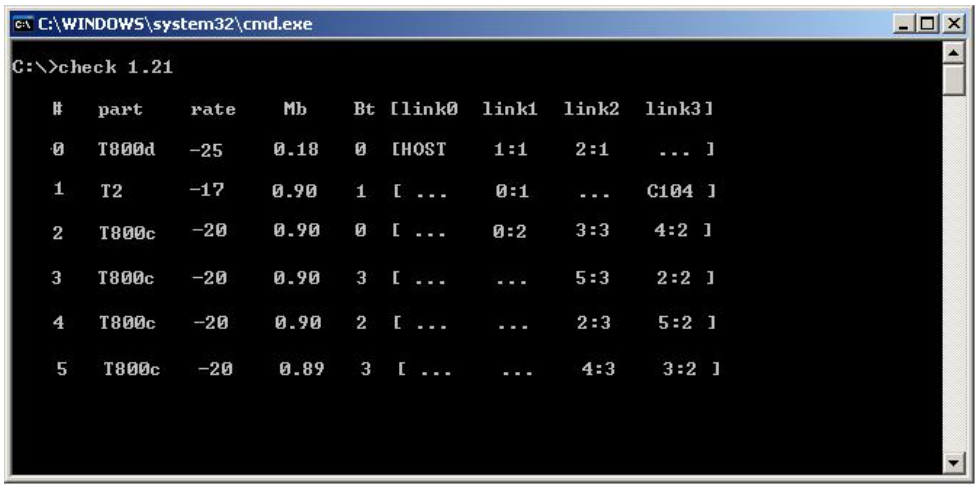
ดังนั้นคำสั่งทั้งหลายที่จะนำมาใช้ในการ โปรแกรม I/O port เป็นคำสั่งที่เกี่ยวกับ Memory Load และ Store ทั้งหมดทั้งสิ้น สิ่งที่ทำให้คำสั่งเหล่านี้เข้าไปถึง Link นั้นได้ก็คือจะใช้แอดเดรสที่ Map กับ Link Register แต่จะเป็นอะไรถ้าแอดเดรสมา Map กับ Physical Memory ? คำตอบก็คือว่า ๒ โพรเซสที่ Run อยู่บน Transputer ตัวเดียวกันทั้ง ๒ งานใช้แอดเดรสเดียวกัน แต่ว่าโพรเซสหนึ่งเป็นตัวที่ถูกลงตำแหน่งแรกของไบต์ที่จะส่งออก และอีกตัวก็โหลดไบต์แรกของข้อมูลที่เข้ามา Transputer จึงสามารถทำงานได้พร้อมกัน

นี่เป็นวิธีการที่อัจฉริยะของบริษัท Inmos ที่ทำลายอุปสรรคทางกายภาพที่เกิดขึ้นในขณะที่ทำการกระจายข้อมูลที่เป็นงานที่อยู่บน Transputer ตัวเดียวกัน หรือบน Transputer ตัวที่อยู่ใกล้เคียงกันมันสามารถที่จะรับส่งข้อมูลโดยใช้ซอฟต์แวร์อันเดียวกันจริง ๆ ได้โดยอิสระ ไม่ว่าจะใช้ตัวกลางในการส่งข้อมูลใดก็ตาม ไม่ว่าจะเป็น Physical Link ตัวกลางที่เหมือนกันในหน่วยความจำตำแหน่งนั้น ๆ เราจึงสามารถแยกความแตกต่างระหว่าง Hard Channel (การรับส่งข้อมูลของตัวที่อยู่ห่างกัน) และ Soft Channel (เป็นการรับส่งข้อมูลระหว่างโพรเซส ใน Transputer ตัวเดียวกัน)



รูปที่ ๙ การสื่อสารระหว่าง Transputer ผ่านทาง Channel ต่าง ๆ

เครื่องมือ Check ๑.๒๑ เป็นคำสั่งที่ช่วยในการตรวจสอบการเชื่อมต่อ Channel ต่าง ๆ ของ Transputer โดยเมื่อต่อเครือข่าย Transputer เข้ากับเครื่องคอมพิวเตอร์แล้วสามารถตรวจสอบได้ดังนี้



ดังนั้น ถ้าเรารู้ Microcode เราจะสามารถรู้ความแตกต่างของ Hard Channel และ Soft Channel ได้ คำสั่งของภาษาแอสเซมบลี (Assembly) ไม่สามารถที่จะบอกความแตกต่างได้และด้วยเหตุผลนี้จึงไม่มี Code ในโปรแกรมที่ใช้ภาษาระดับสูง ซึ่งหมายความว่า โปรแกรมคู่ขนานหนึ่ง ๆ จะบรรจุด้วยโพรเซสหลาย ๆ โพรเซสในเวลาเดียวกัน สามารถที่จะนำมาดัดแปลงใช้โดยมีการเปลี่ยนแปลงเพียงเล็กน้อยที่จะใช้ Run ในระบบ Multi Transputer ระบบหนึ่งๆ ได้ ซึ่งโพรเซสต่าง ๆ จะถูกกระจายออกไปยัง Transputer หลาย ๆ ตัว และสามารถที่จะรับส่งข้อมูลซึ่งกันและกันโดยผ่าน Channel ต่าง ๆ เกิดขึ้นได้ในวิธีที่ว่า Physical Location ของงานชนิดที่สัมพันธ์ซึ่งกันและกัน