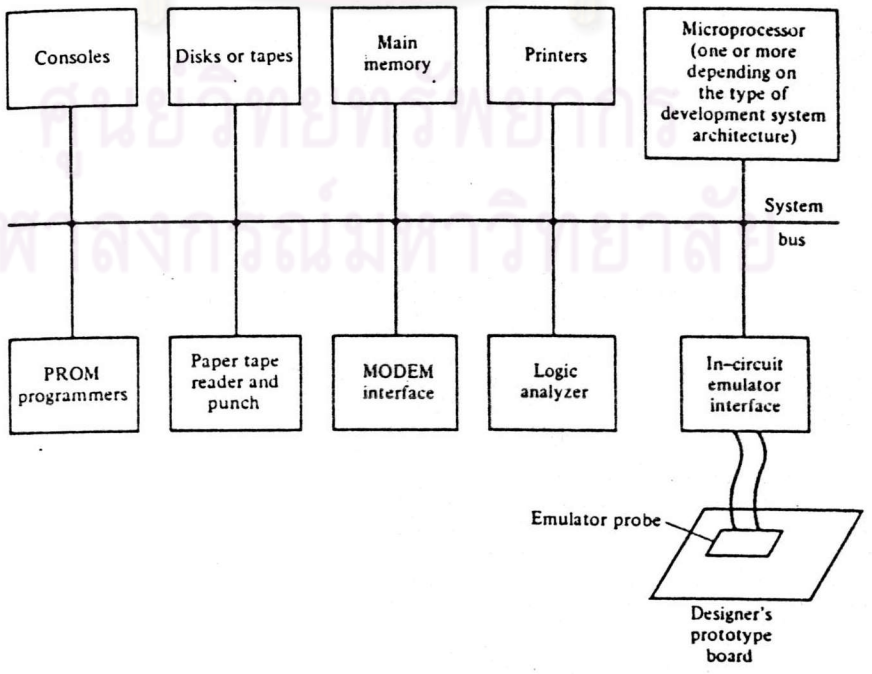
บทที่ 2

ระบบพัฒนาไมโครโปรเซสเซอร์ (MDS)

2.1 MDS คืออะไร [2]

ระบบพัฒนาไมโครโปรเซสเซอร์หรือ Microprocessor Development System (MDS) เป็นระบบที่จำเป็นในการออกแบบและพัฒนาระบบที่ใช้ไมโครโปรเซสเซอร์ ยิ่งระบบไมโครโปรเซสเซอร์ที่จะออกแบบยุ่งยากมากเท่าไร ความจำเป็นที่ต้องมีระบบช่วยในการพัฒนาซอฟต์แวร์และเทคนิคในการดีบักฮาร์ดแวร์ก็ยิ่งมีมากขึ้นเท่านั้น ระบบพัฒนาไมโครโปรเซสเซอร์ที่ใช้กันจะมีตั้งแต่ขนาดเล็ก ราคาไม่แพงนัก และมีขีดความสามารถจำกัดซึ่งใช้พัฒนาได้เฉพาะทางด้านซอฟต์แวร์ จนถึงระบบพัฒนาไมโครโปรเซสเซอร์ขนาดใหญ่ที่สามารถสนับสนุนการพัฒนาได้ทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์จากผู้ใช้หลาย ๆ คนในขณะเวลาเดียวกันได้ ผู้ที่สร้างระบบพัฒนาไมโครโปรเซสเซอร์จะมีอยู่ 2 กลุ่ม คือ กลุ่มผู้ผลิตอุปกรณ์ และกลุ่มผู้ที่ใช้อุปกรณ์ ระบบพัฒนาที่สร้างจากกลุ่มผู้ผลิตไมโครโปรเซสเซอร์ จะใช้ได้เฉพาะการพัฒนาระบบที่ใช้ไมโครโปรเซสเซอร์เบอร์นั้น ๆ (Nonuniversal System) ส่วนระบบพัฒนาที่สร้างจากกลุ่มที่ไม่ใช่ผู้ผลิตหรือกลุ่มผู้ใช้ จะสามารถรองรับการพัฒนาระบบได้หลายเบอร์ (Universal Systems) ระบบพัฒนาไมโครโปรเซสเซอร์หนึ่ง ๆ ประกอบด้วยฮาร์ดแวร์ที่จำเป็นในการออกแบบระบบไมโครโปรเซสเซอร์ และชุดของโปรแกรมหรือซอฟต์แวร์ที่จำเป็นในการทำงานของฮาร์ดแวร์ ระบบพัฒนาไมโครโปรเซสเซอร์ส่วนที่เป็นฮาร์ดแวร์แสดงได้ดังรูปที่ 2.1 ได้แก่



รูปที่ 2.1 แสดงองค์ประกอบที่เป็นฮาร์ดแวร์ของระบบพัฒนาไมโครโปรเซสเซอร์

1. ไมโครโปรเซสเซอร์ (Microprocessor) จำนวนไมโครโปรเซสเซอร์ที่ใช้ในระบบพัฒนาจะขึ้นอยู่กับสถาปัตยกรรมของระบบ ระบบที่มีตัวประมวลผลหลายตัวสามารถทำงานได้หลายอย่างและมีประสิทธิผลสูงแต่ส่วนใหญ่ระบบพัฒนาที่เป็นระบบ Single-station มักจะมีตัวประมวลผลตัวเดียวซึ่งเพียงพอสำหรับการพัฒนาระบบเล็ก ๆ

2. หน่วยความจำหลัก (Main Memory) เป็นหน่วยความจำที่ตัวประมวลผลของระบบพัฒนาเข้าถึงได้โดยตรงซึ่งจะใช้เก็บซอฟต์แวร์ของการพัฒนาเช่น เอดิเตอร์ (Editor) แอสเซมเบลอร์ (Assembler) และอีมูเลเตอร์ (Emulator) เป็นต้น หน่วยความจำหลักมักมีขนาดตั้งแต่ 32 K จนถึง 64 K และประกอบด้วยรอม (ROM) และแรม (RAM) โดยปกติรอมจะมีโปรแกรมมอนิเตอร์ (Monitor) ซึ่งใช้ในการอินเตอร์เฟส ซอฟต์แวร์ของระบบกับฮาร์ดแวร์ของระบบ

3. หน่วยเก็บข้อมูล (Mass Storage) ซึ่งได้แก่ ดิสก์และเทปแม่เหล็ก เป็นตัวเก็บซอฟต์แวร์ของระบบพัฒนาและโปรแกรมของผู้ใช้เมื่อไม่ได้ใช้งาน ตัวประมวลผลของระบบพัฒนาไม่สามารถเข้าถึงหน่วยเก็บข้อมูลนี้ได้โดยตรงแต่ต้องติดต่อผ่านทางอุปกรณ์ไอ/โอ ต่าง ๆ ซอฟต์แวร์ OS (Operating System Software) จะมีโปรแกรมย่อยที่ใช้อินเตอร์เฟสระหว่างผู้ใช้กับหน่วยเก็บข้อมูล โดยจะโหลดโปรแกรมจากหน่วยเก็บข้อมูลไปไว้ที่หน่วยความจำหลักก่อน

4. CRT คอนโซล (CRT console) เป็นตัวที่ใช้ในการอินเตอร์เฟสระหว่างผู้ใช้กับ OS โดยผู้ใช้ป้อนอินพุตผ่านทางคีย์บอร์ด (CRT Keyboard) และ OS แสดงข้อมูลบนจอ (CRT Screen) ข้อมูลที่แสดงบนจอได้แก่ สถานะของตัวประมวลผล, โปรแกรมที่ดีบักและข้อความที่แสดงข้อผิดพลาดในระหว่างการดีบัก เป็นต้น ปกติ CRT จะเชื่อมต่อกับระบบพัฒนาแบบการรับส่งอนุกรม ซึ่งมีอัตราเร็วโดยเฉลี่ยประมาณ 9,600-19,200 บิตต่อวินาที

5. อินเซอร์กิตอีมูเลเตอร์ (In-Circuit Emulator : ICE) ระบบพัฒนาส่วนใหญ่จะมี ICE อย่างน้อยหนึ่งตัว ICE เป็นเครื่องมือที่ก้าวหน้ามากที่สุดสำหรับการพัฒนาฮาร์ดแวร์ของไมโครโปรเซสเซอร์ให้เป็นเครื่องมือทดสอบ การใช้งานอีมูเลเตอร์จะถอดเอาตัวไมโครโปรเซสเซอร์ของระบบที่จะพัฒนาออก (Target Processor) และนำเอาสายต่อจากอีมูเลเตอร์ไปเสียบแทนที่ อีมูเลเตอร์จะทำงานเลียนแบบการทำงานของตัวไมโครโปรเซสเซอร์ที่ถอดออกไปภายใต้การควบคุมของซอฟต์แวร์ของระบบพัฒนา อีมูเลเตอร์สามารถทำงานร่วมกับฮาร์ดแวร์ของระบบที่จะพัฒนา (Target System)และแสดงข้อมูลสถานะต่าง ๆ เกี่ยวกับขบวนการทำงานได้ ทั้งยังสั่งให้ทำงานทีละคำสั่งได้อีกด้วย ผู้ใช้ ICE สามารถหาบักของฮาร์ดแวร์และซอฟต์แวร์ได้อย่างรวดเร็ว ซึ่งอาจต้องใช้เวลาหลายชั่วโมงหากใช้อุปกรณ์

ทดสอบอื่น ๆ

6. เครื่องวิเคราะห์ลอจิก (Logic Analyzer) เป็นเครื่องมือในการดีบักฮาร์ดแวร์ที่ดีมากอีกอย่างหนึ่ง ซึ่งผู้ใช้สามารถดูผลของระดับลอจิกที่จะเกิดขึ้นในการทำงานจริงของฮาร์ดแวร์ได้ เครื่องวิเคราะห์ลอจิกยังสามารถตรวจจับและแสดงสัญญาณรบกวนต่างๆ ที่เป็นสาเหตุให้ฮาร์ดแวร์ไม่ทำงานด้วยโดยจะสุ่มสัญญาณอินพุตซึ่งปกติมี 16 ถึง 32 เส้นที่จังหวะสัญญาณนาฬิกาต่าง ๆ และเก็บเป็นค่าไบนารี (Binary) ในหน่วยความจำ เพื่อจะนำมาแสดงเป็นลำดับของสัญญาณ "1" และ "0" บนจอ CRT

7. เครื่องโปรแกรมอีพรอม (EPROM Programmer) หลังจากพัฒนาและดีบักซอฟต์แวร์ของ Target แล้วก็จำเป็นต้องมีที่เก็บโปรแกรมเพื่อใช้ระบบที่พัฒนาขึ้นมาในงานจริง เครื่องโปรแกรมอีพรอมจะนำเอาโปรแกรมภาษาเครื่องไปโปรแกรมลงในอีพรอม

8. เครื่องพิมพ์ (Line Printer) โปรแกรมที่พัฒนาและข้อมูลต่าง ๆ จะถูกพิมพ์ออกมาทางเครื่องพิมพ์ เพื่อการเรียกดูและเก็บเป็นข่าวสารข้อมูล

9. ตัวอินเตอร์เฟสกับเทปกระดาษ (Paper Tape Interfaces) บางครั้งการติดต่อสื่อสารระหว่างระบบที่ต่างกันต้องอาศัยเทปกระดาษที่เจาะรูบนระบบหนึ่ง แล้วอ่านผ่านตัวอ่านเทปกระดาษเพื่อเอาข้อมูลเข้าอีกระบบหนึ่ง แต่วิธีนี้เลิกใช้งานแล้วเพราะช้าและไม่สะดวกในการเรียกใช้

10. ตัวอินเตอร์เฟสโมเด็ม (MODEM Interfaces) โมเด็ม (MOdulator-DEModulator) เป็นตัวติดต่อสื่อสารแบบอนุกรมผ่านทางสายโทรศัพท์ โดยตัวโมดูเลเตอร์ทางด้านตัวส่งจะแปลงข้อมูลแบบอนุกรมเป็นสัญญาณเสียงผ่านทางสายโทรศัพท์ และตัวดีโมดูเลเตอร์ทางด้านตัวรับจะรับสัญญาณแล้วแปลงกลับมาเป็นข้อมูลแบบอนุกรม นอกจากนี้โมเด็มยังใช้ติดต่อระหว่าง CRT คอนโซลของผู้ใช้หนึ่ง ๆ กับระบบพัฒนาส่วนกลางได้อีกด้วย อัตราเร็วในการรับส่งข้อมูลสำหรับโมเด็มจะถูกจำกัดอยู่ที่ 1,200 บิตต่อวินาที ซึ่งช้ากว่าการติดต่อของ CRT ระบบพัฒนาไมโครคอมพิวเตอร์สามารถเชื่อมต่อกับมินิคอมพิวเตอร์ผ่านทางโมเด็มเพื่อเพิ่มความสามารถในการพัฒนาซอฟต์แวร์สำหรับไมโครโปรเซสเซอร์ต่าง ๆ กันด้วย

ส่วนซอฟต์แวร์ของระบบพัฒนาไมโครโปรเซสเซอร์ ได้แก่

1. OS (Operating System) เป็นตัวจัดการเกี่ยวกับซอฟต์แวร์ที่ใช้พัฒนาระบบไมโครโปรเซสเซอร์โดยจะติดต่อกับฮาร์ดแวร์ของระบบผ่านทางโปรแกรมย่อยต่าง ๆ OS จะประกอบด้วยโปรแกรมที่จัดการเกี่ยวกับหน่วยความจำ โปรแกรมที่ติดต่อกับผู้ใช้และโปรแกรมช่วยต่างๆ(Utility Program) ที่ให้ผู้ใช้สามารถจัดการเกี่ยวกับไฟล์ (file) ได้ เช่น สร้างไฟล์ ลบไฟล์ ดูพลิเคทไฟล์ หรือไดเรกตอรีไฟล์

2. เอดิเตอร์ (Editor) เป็นโปรแกรมแรกที่ใช้ในการพัฒนาซอฟต์แวร์ ผู้ใช้จะป้อน Source code ซึ่งอาจเขียนด้วยภาษาแอสเซมบลีหรือภาษาชั้นสูงอื่น ๆ ไปยังระบบที่จะพัฒนาโดยผ่านทางเอดิเตอร์ เอดิเตอร์จะมีคำสั่งสำหรับเพิ่มเติมหรือลบแก้ไขโปรแกรมด้วย

3. แอสเซมเบลอร์ (Assembler) และคอมไพล์เลอร์ (Compiler) เป็นโปรแกรมที่แปลง Source code จากเอดิเตอร์ ซึ่งถูกเก็บไว้ในรูปแบบของตัวอักษร ASCII มาเป็น Object code หรือ Machine code ซึ่งเก็บในลักษณะเป็น Relocatable file

4. ลิงค์เกอร์ (Linker) เป็นโปรแกรมที่เปลี่ยน Object file จากแอสเซมเบลอร์ซึ่งเป็น Relocatable file ให้เป็น Absolute file ซึ่งมี Machine code ที่กำหนดแอดเดรสแน่นอนแล้ว

5. โหลดเดอร์ (Loader) เป็นโปรแกรมที่นำเอา Absolute file มาโหลดลงสู่หน่วยความจำของระบบที่ต้องการจะพัฒนา

6. ดีบักเกอร์ (Debugger) เป็นโปรแกรมที่ทำให้ผู้ใช้สามารถสั่งรันโปรแกรมภายใต้การควบคุมเงื่อนไขต่าง ๆ และให้ผู้ใช้สามารถควบคุมแบบจำลองของเครื่องต้นแบบได้ โดยทั่วไปจะมีความสามารถ หรือ Tools ที่ช่วยในการดีบักซอฟต์แวร์ ดังนี้

- Single Step Facility เป็นความสามารถในการทำงานทีละคำสั่งเพื่อติดตามขั้นตอนการทำงานของโปรแกรมอย่างใกล้ชิด

- Breakpoint Facility เป็นความสามารถในการหยุดทำงาน ณ จุดที่กำหนดเพื่อให้สามารถทดลองรันโปรแกรมที่ความเร็วจริงแล้วหยุดเพื่อตรวจสอบค่าต่าง ๆ ได้

- Register Dump Program มีโปรแกรมส่วนที่ให้ผู้ใช้สามารถตรวจสอบแก้ไขค่าในรีจีสเตอร์ผ่านทาง CRT คอนโซล

- Memory Dump Program มีโปรแกรมส่วนที่ให้ผู้ใช้สามารถตรวจสอบแก้ไขค่าในหน่วยความจำผ่านทาง CRT คอนโซล

- Simulator Program มีโปรแกรมส่วนที่ให้ผู้ใช้สามารถทดสอบรันโปรแกรมและเก็บค่าพารามิเตอร์ต่าง ๆ ไว้เช่น ค่าแอดเดรส ข้อมูลหรือแฟลก(flag)เป็นต้น

7. มอนิเตอร์ (Monitor) เป็นโปรแกรมที่อำนวยการทำงานทั้งหมดของระบบพัฒนาโดยมีโปรแกรมย่อยที่เป็นตัวเชื่อมต่อระหว่างโปรแกรมต่าง ๆ กับอุปกรณ์ไอ/โอในระบบ

เครื่องมือทั้งหมดที่จำเป็นในการออกแบบและพัฒนาระบบไมโครโปรเซสเซอร์ จะถูกจัดรวมเป็นชุดของระบบพัฒนาไมโครโปรเซสเซอร์ ชุดของระบบพัฒนาไมโครโปรเซสเซอร์ที่มีใช้กันอย่างแพร่หลาย ได้แก่ ระบบพัฒนาไมโครโปรเซสเซอร์ของบริษัท อินเทล (Intel120 ,225,286 และ 290) ระบบพัฒนาไมโครโปรเซสเซอร์ของบริษัทเทกทรอนิกส์(Tektronix

8540, 8550, 8560 และ 8002A) และระบบพัฒนาไมโครโปรเซสเซอร์ของบริษัทฮิวเล็ทแพกการ์ด (Hewllett Packard 64000) เป็นต้น ซึ่งเราสามารถเปรียบเทียบฟังก์ชันต่าง ๆ ของแต่ละระบบได้ในตารางที่ 2.1 [2]

Development System Emulator Architecture and Associated Hardware/Software Devices

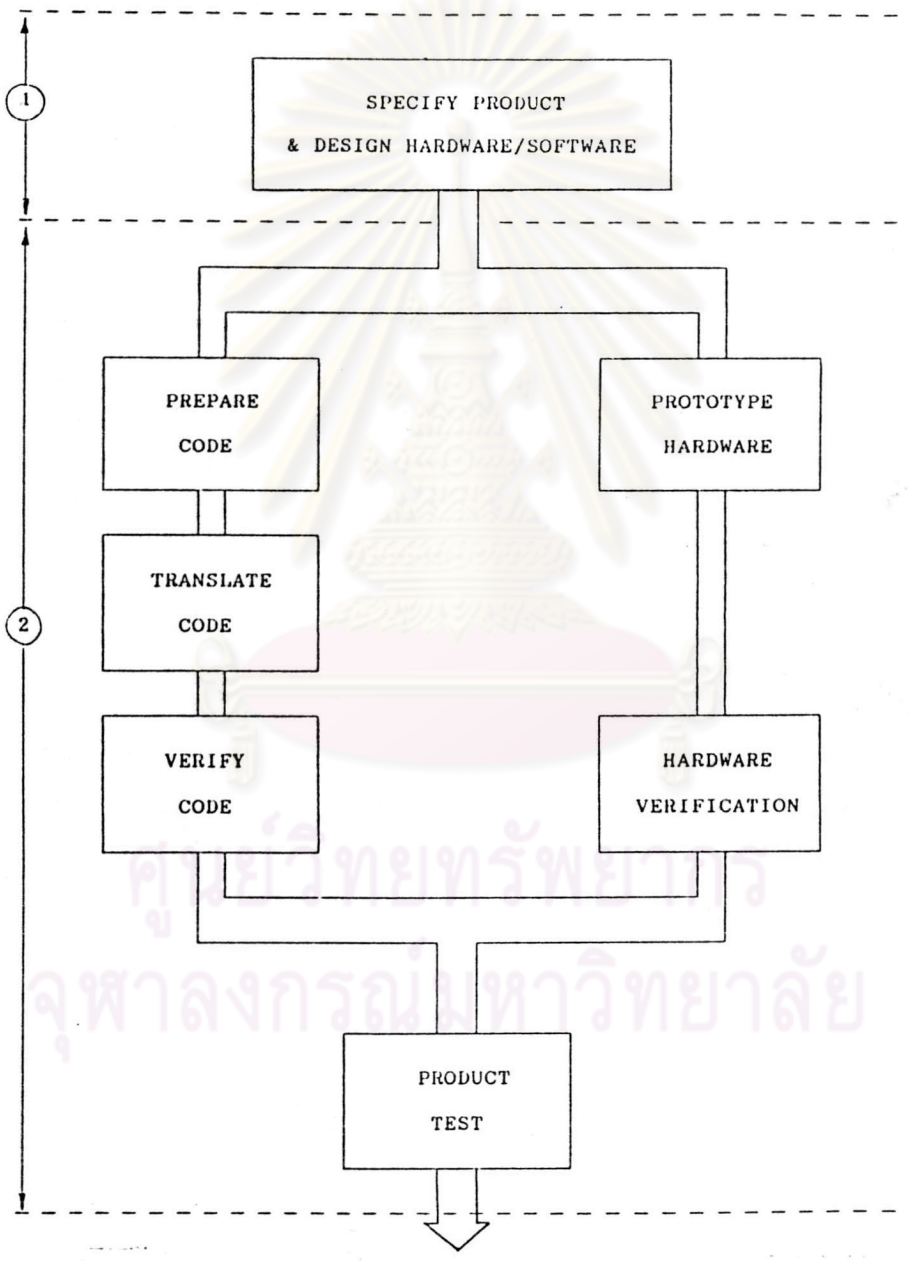
MANUFACTURER	CHARACTERISTICS	MICROPROCESSORS	MEMORY AND PERIPHERALS	SOFTWARE	CRT DISPLAY
Hewlett-Packard 64000	Master/slave multistation with (up to 6) complete separations of host processor from the emulation system. This allows the host processor to run the emulation support software independently of the emulator Universal Soft key features Provides both hardware and software development capabilities	8080 8085 8086 6800 6802 68000 8048 8049 6809 8040 8035 8039 8050	HP 16-bit host microprocessor with 128K of 16-bit host and 64K bytes of emulation memory Hard disk of up to 960 megabytes Line printers Universal PROM programmers Logic analysis Cassette interface	Editor Assembler Macroassembler Linker PASCAL compiler	Memory refreshed (smart terminal)
Intel 120	Basically similar to Intel's previous model 220 Modified master/slave single memory Nonuniversal In-circuit emulator is provided as an option	8080/8085 8048/8049 May expand to support other Intel microprocessors	8080 microprocessor host with 32K bytes of RAM and 4K bytes of ROM 250K byte floppy disk Universal PROM programmers High-speed paper tape reader/punch Single station Line printer	Editor Assembler Macroassembler Debugger Linker No high-level language compilers of interpreters in the basic system	Memory refreshed (smart terminal)
Intel 225	Similar to previous Model 230. The main difference is 230 was 8080 based whereas 225 is 8085 based. Modified master/slave Nonuniversal Standard emulation capabilities	8080/8085 8048/8049 8086	8085 host microprocessor with 64K bytes of RAM and 4K bytes ROM 250K bytes floppy disk plus 7.3 megabytes hard disk Single station PROM programmer Line printer Tape reader/punch	Editor Assembler Macroassembler Debugger Linker PL/M compiler FORTRAN compiler PASCAL compiler COBOL compiler	Memory refreshed (smart terminal)
Intel 286	A new system with two host processors (8085 and 8086). 8086 host supports development of 8080/8085 and 8048. Nonuniversal Modified master/slave Standard emulation capabilities	8086/8088 8080/8085 8048	Two host processors with 96K bytes of user RAM 250K bytes floppy disc and 7.3 megabytes of hard disk Single Station Paper tape reader/punch PROM programmer Line printer	Editor Assembler Macroassembler Debugger Linker Supports PL/M 88/86 PASCAL 88/86 FORTRAN 88/86	Memory refreshed (smart terminal)
Intel 290	A new network with 8085 host and supporting up to 8 stations Nonuniversal Modified master/slave Supports all Intel development systems as work stations	8080/8085 8048/8049 8086/8088	8085 host with up to 8 stations 64K RAM, 4K ROM Maximum 15 megabytes of hard and floppy disks Line printer PROM programmer	Editor Assembler Macroassembler Debugger Linker Supports PL/M 88/86 PASCAL 88/86 FORTRAN 88/86	Memory refreshed (smart terminal)
Tektronix 8002A	Master/slave with split memory Provides both hardware and software development capabilities Three emulation modes Universal	Supports 29 microprocessors such as: 8086 68000 Z8000 8080/8085 6800, 6802 1302 and many more	16K byte of host and up to 64K byte of target Floppy disk PROM programmer Logic analyzer Printer Single station	Editor Assembler Debugger Macroassembler Linker	RS-232C (dumb terminal)
Tektronix 8550	Single-user system Universal Master/slave Can be interfaced to multiuser systems	29 microprocessors	Two hardware units plus optional CRT terminal and line printer Dual floppy disk Logic analyzer	Editor Assembler Debugger Linker Macroassembler MDL/μ and PASCAL compilers	RS-232C (dumb terminal)
Tektronix 8560	Time-shared software development Multiuser up to 8 stations Uses DEC LSI 11/23 8001, 8002A, and 8550 can be added to provide hardware development stations Universal	29 microprocessors	128K words dynamic RAM 35 megabytes hard disk 1 megabyte floppy disk Line printer PROM programmer	Editor Assembler Debugger Linker Macroassembler PASCAL compilers	RS-232C (dumb terminal)
Tektronix 8540	Provides multiuser capabilities via a host Similar to 8560 with emulation capabilities Master/slave Expandable using 8002A Universal Developed basically for host system owners	29 microprocessors	32K byte RAM PROM programmer Logic analyzer Depends on host	Debugger System development software depends on host	RS-232C (dumb terminal)

2.2 ขั้นตอนการพัฒนาระบบไมโครโปรเซสเซอร์

การออกแบบและพัฒนาระบบไมโครโปรเซสเซอร์โดยทั่วไป แบ่งออกได้ 2 ช่วงคือ

1. ช่วงการออกแบบ เป็นช่วงที่ตั้งข้อกำหนดและกำหนดแนวทางการพัฒนาระบบ

2. ช่วงการทำ เป็นช่วงออกแบบประกอบวงจรและเขียนโปรแกรม ซึ่งมีการทดสอบควบคู่ไปด้วยและเมื่อพัฒนาทั้งฮาร์ดแวร์และซอฟต์แวร์จนเรียบร้อยแล้ว จึงนำมาทดสอบการทำงานร่วมกันอีกครั้ง



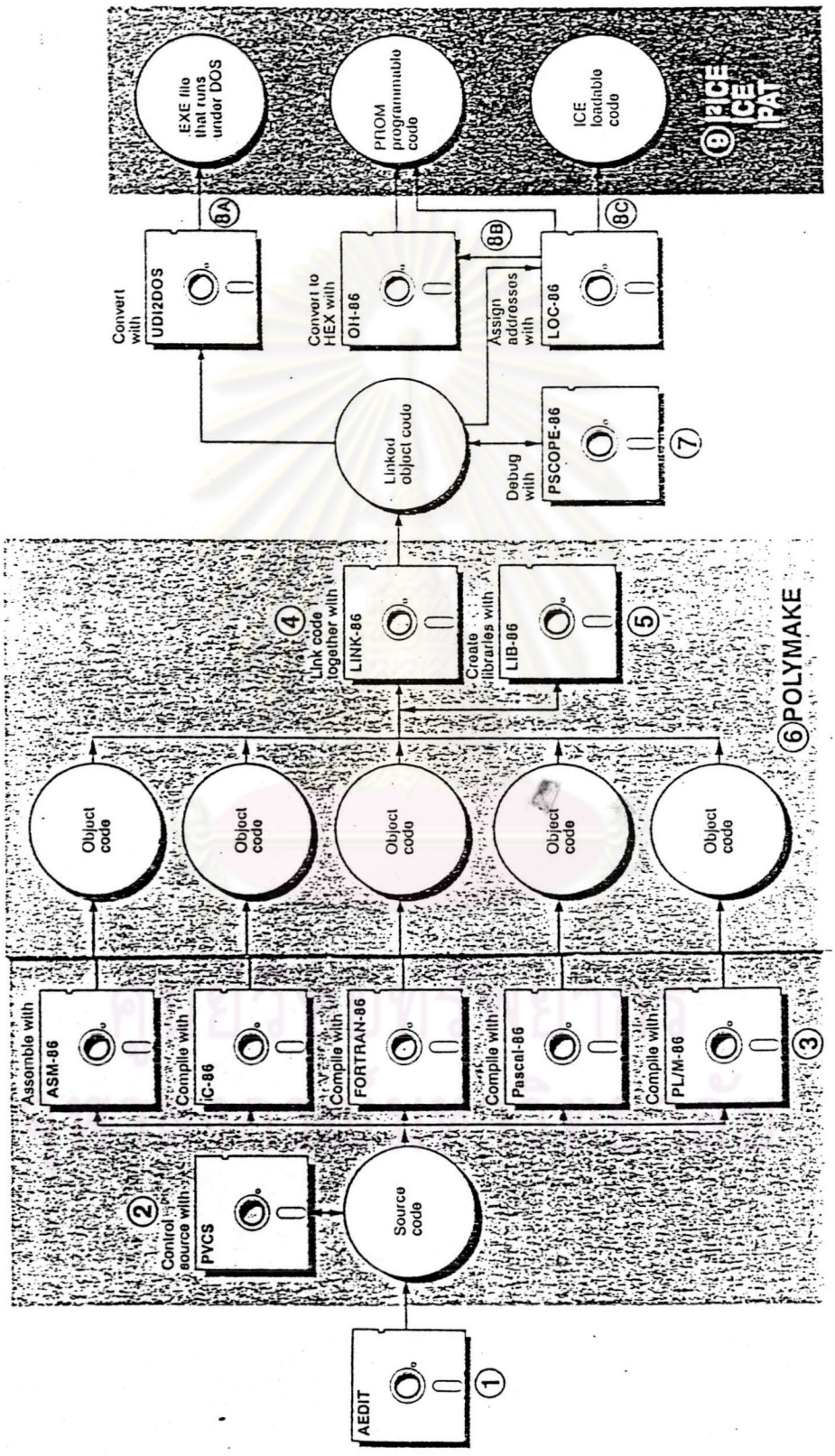
รูปที่ 2.2 แสดงขั้นตอนการออกแบบและพัฒนาระบบไมโครโปรเซสเซอร์

จากรูปที่ 2.2 การออกแบบสร้างและทดสอบฮาร์ดแวร์ซอฟต์แวร์จะเริ่มทำแยกกันก่อนโดยหลังจากที่ได้กำหนดแนวทางการออกแบบแล้ว ผู้ออกแบบฮาร์ดแวร์จะออกแบบและ

ประกอบวงจรขึ้นมาเป็นเครื่องต้นแบบ จากนั้นก็ต้องเขียนโปรแกรมทดสอบเฉพาะระบบนั้น ๆ และใช้เครื่องมือตรวจจับสัญญาณ เช่น ออสซิโลสโคป (Oscilloscope) หรือลอจิกโพรบ (Logic Probe) ในการตรวจหาความผิดพลาดของวงจร ส่วนผู้ออกแบบซอฟต์แวร์จะเริ่มจากการออกแบบเขียนแผนผังการทำงานของโปรแกรม (Flowchart) เตรียมข้อมูลเพื่อป้อนโปรแกรมเข้าสู่เอดิเตอร์ซึ่งโปรแกรมอาจจะเขียนด้วยภาษาแอสเซมบลี หรือภาษาชั้นสูงอื่น ๆ จากนั้นก็ทำการแปลโปรแกรมด้วยแอสเซมเบลอร์หรือคอมไพล์เลอร์ แล้วผ่านการลิงค์เพื่อให้ได้เป็นโปรแกรมภาษาเครื่องที่ถูกกำหนดแอดเดรสแน่นอน ต่อไปจึงนำโปรแกรมที่ได้มาตรวจสอบลำดับการทำงานภายใต้สภาวะที่ใกล้เคียงกับความเป็นจริงมากที่สุด ซึ่งอาจจะใช้ดีบักเกอร์ ซีมูเลเตอร์ หรืออีมูเลเตอร์ก็ได้ หลังจากแก้ไขโปรแกรมเรียบร้อยแล้ว จึงนำโปรแกรมมาทดสอบกับเครื่องต้นแบบที่สร้างขึ้นเพื่อตรวจสอบและแก้ไขอีกครั้ง โปรแกรมที่ได้ดีบักจนแน่ใจแล้วจะนำไปโปรแกรมลงอีพรอมเพื่อใช้ในงานจริง

จากที่กล่าวมา แต่ละขั้นตอนของการออกแบบและพัฒนาระบบไมโครโปรเซสเซอร์จำเป็นต้องมีชุดพัฒนาระบบไมโครโปรเซสเซอร์เพื่อตรวจสอบแก้ไข ทั้งส่วนของฮาร์ดแวร์และซอฟต์แวร์ ชุดพัฒนาระบบไมโครโปรเซสเซอร์มีทั้งแบบที่ใช้บนเครื่องที่สร้างขึ้นโดยเฉพาะแบบที่ใช้บนเครื่องมินิคอมพิวเตอร์ และแบบที่ใช้บนเครื่องไมโครคอมพิวเตอร์ ชุดพัฒนาระบบไมโครโปรเซสเซอร์ที่ใช้บนเครื่องไมโครคอมพิวเตอร์ยี่ห้อ IBM PC ก็มีด้วยกันหลายระบบ แต่จะขอยกตัวอย่างชุดพัฒนาของบริษัทอินเทล ที่ใช้บนเครื่องไมโครคอมพิวเตอร์ IBM PC ชุดหนึ่งซึ่งสามารถชี้ให้เห็นขั้นตอนการพัฒนาโปรแกรมเป็นลำดับไป ในตัวอย่างนี้เป็นชุดพัฒนาของระบบที่ใช้ไมโครโปรเซสเซอร์ 8086 ดังรูปที่ 2.3

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.3 แสดงชุดพัฒนาระบบไมโครโปรเซสเซอร์ 8086 ของอินเทล

1. ขั้นแรกเริ่มจากการป้อนโปรแกรมที่ได้ออกแบบไว้ ผ่านทางคีย์บอร์ดของเครื่องไมโครคอมพิวเตอร์ IBM PC เข้าสู่โปรแกรม AEDIT เพื่อสร้าง Source Code

2. ใช้โปรแกรม PVCS ในการควบคุมการแก้ไข Source Code

3. เลือกใช้แอสเซมเบลอร์หรือคอมไพล์เลอร์ที่เหมาะสมกับ Source Code ซึ่งในชุดพัฒนาของอินเทล มีคอมไพล์เลอร์สำหรับภาษา C ภาษาฟอร์แทรน ภาษาปาสคาลและภาษา PL/M

4. หลังจากได้ Object Code จากขั้นตอนที่ 3 นำมาผ่านการลิงค์โดยโปรแกรม LINK-86

5. ผู้ใช้อาจต้องการสร้างโมดูลจาก Object Code เป็น Library ไว้ใช้ โดยผ่านโปรแกรม LIB-86

6. โปรแกรม Poly make ใช้กับโปรแกรม PVCS อีกที

7. ทดสอบและดีบักโปรแกรมในระดับภาษาชั้นสูงด้วยโปรมแกรม Pscode-86 ซึ่งเป็นดีบักเกอร์ของชุดพัฒนานี้

8. จัดการเปลี่ยนโปรแกรมที่จะพัฒนาไปเป็นรูปแบบที่ต้องการโหลดลงหน่วยความจำของระบบเพื่อทดสอบครั้งสุดท้าย

9. ทำการดีบักโปรแกรมด้วยอีมูเลเตอร์ I^2ICE, ICE หรือ iPAT (intel's Performance Analysis Tool)

หลังจากขั้นตอนที่ 9 ผู้ใช้จะได้โปรแกรมที่พัฒนาเรียบร้อยแล้ว และนำไปใช้ในงานจริงได้

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย