Interoffice Correspondence

□ For Your Comments
□ For Your Approval
□ Please Take Action
□ Call Me
□ See Me
□ Per Our Conversation
□ Read, Initial & Return

□ For Your Information
□ Note & Return
□ Note & File
□ For Your Signature
□ Please Advise Status
□ Per Your Request
□ Please Advise Appropriate Employees In Your Department

To: Mike Holley, Russ dePina, George James, Wayne Paulson, Charles Golm, Bill To

Date: April 22, 1982

CC:

From: Vic Belmondo

Subject: High Level Logic Language

Please plan to attend a meeting on Monday, April 26 at 10:00 in conference room 2D to discuss the following items:

10 15

1. Current status of ABEL development	10 minutes
2. Relationship of ABEL to current Logic system plan	20 minutes
3. Relationship of ABEL to future long range plans	20 minutes
4. Develop action items	10 minutes

VEB

VEB:lc
Att.

Interoffice Correspondence

□ For Your Comments
□ For Your Approval
□ Please Take Action
□ Call Me
□ See Me
□ Per Our Conversation
□ Read, Initial & Return

□ For Your Information
□ Note & Return
□ Note & File
□ For Your Signature
□ Please Advise Status
□ Per Your Request
□ Please Advise Appropriate Employees In Your Department

To: Greg Tannheimer
Mike Mraz
Wayne Paulson

Date: April 13, 1982

CC: All P.D.C. Members

From: Vic Belmondo

Subject: Black Box Approach to Logic Programming

DATA I/O's most important programmer market question is to define a role in the emerging Logic market. An aggressive leadership role is outlined below.

In hardware, provide the computing power for complete design, universal programming capability and universal testing capability.

In software and application support, parallel microprocessor development systems support. Emphasize support from cradle to grave - from design to programming to testing. This can be accomplished by taking a black box approach. The designer does not have to know what is in the logic device, only what he wants the device to do. This is how microprocessors are currently viewed.

We can avoid the problems that plagued microprocessor system development for years and still plague us at Data I/O. These problems are design time, maintainability, and transportability. High level development languages are being used to solve these problems.

Problem	Benefit of High Level Development Language
Design Time	Fewer lines of code = faster design, less debug time
Maintainability	Self-documenting = uniform documentation, no tricks are designed in
Transportability	Processor independent = software can be re-used.

The penalty usually is an increase in system memory and possibly system operating speed for high level language implementations as compared to assembly languages.

An exact parallel can be drawn for design of systems using programmable logic.

Problem	Benefit of Black Box Approach
Design time	Function to fuses automatically.
Maintainability	No tricks in device - imagine troubleshooting a system with 20 tricky FPLs or PALs. Automatic and consistent documentation.
Transportability	Once function is expressed in high level language it can be implemented in PALs, FPLs, etc., depending on compression desired or availability of devices.

The penalty of using a black box approach could be less efficient use of logic devices.

Two levels of language could be developed. A Boolean language like PALASM is equivalent to assembly language. Another "assembly" level language for sequential machines might be like the following example:

Current state: $\underline{1}$ $\underline{3}$
Current output: $\underline{X}$ $\underline{Y}$ $\underline{Z}$
Next input: $\underline{A}$ $\underline{B}$ $\underline{C}$
Next state: $\underline{1}$ $\underline{4}$
Next output: $\underline{X}$ $\underline{Y}$ $\underline{Z}$

A high level language equivalent to PASCAL could be envisioned. It might have commands like Add, Multiply, Divide, Decode, etc.

Automatic software to provide testing algorithms to support the random testing of devices should also be provided.

The devices lend themselves to a black box approach. One Signetics part is a miniature Mealy Machine, a cannonical form to which all sequential machines can be reduced.

Collaboration with the semi houses can enhance our position. All contacts so far indicate that semi houses are extremely willing to support our efforts. John Birkner is collaborating with a university professor to write a textbook on designing with programmable logic. If Data I/O collaborated to write the book, the book and support may be more widely accepted, since it would not have to be slanted to support one kind of logic family.

If Data I/O decides to actively pursue this approach we could sell hardware and/or software. Since Data I/O's basic business is selling hardware, a plan to apply these ideas to the P30 is now presented. High level software and application tools will help us sell hardware and the I.C. houses sell devices.

Current Position:

The P29 with UniPak II, and LogicPak, Gang II will provide the most sophisticated programming capability available for several years. The P22 will provide PROM programming capability until the new generation bipolar PROMs, such as the Fairchild, Signetics and T.I. devices, are widespread.

Proposed P30:

The proposed P30 has as its main attributes a modern case and the lowest possible build cost. These attributes are achieved by eliminating compatibility with existing Paks. However, the P30 must be able to program all or nearly all programmable devices. A P30 UniPak, LogicPak, Gang Module and card sets are proposed to achieve this. As the P30 is now defined, its performance is not differentiated from the P29.

Logic Emphasized P30:

A proposed differentiation of the P30 from P29 is to add capability to address the emerging logic device programming market. This capability would include sufficient number-crunching capability,. interface to a mass storage device (floppy disk) and testing hardware to provide a complete logic device development system.

Reasons to include enhanced logic capability in P30:

1. Capitalize on testing knowledge developed at Data I/O to sell P30 and development software. This supports customer fully for design process. counters customers who only want to buy software for currently owned development systems.

2. Computing power to handle logic is inexpensive and will get more inexpensive. At least 60% of cost of the P30 will be to support programming and testing. Customer pays little for development computing hardware.

3. Gives Data I/O something with which to compete with Stag and other emerging companies concentrating on logic programming.

4. Gives Data I/O leader image.

5. P30 will not compete directlyu with P29 at introduction.

6. Logic support developed first is easily downgraded to PROMs which are a subset of logic. (M.M.I. has developed PALASM tables for 32 x 8 PROMs.) This applies to both hardware and software.

7. Provides product that addresses logic market which is much larger than PROM market if it matures as semi houses predict.

8. P30, since not compatible with current Data I/O paks, competes on equal terms with competitors' equipment.

Reasons for not including enhanced logic capability in P30:

1. Added cost to P30.

2. Risk that resources not available to develop all software required with application and support literature.

3. Risk logic market fails to materialize.

Proposed plan for P30 with enhanced logic:

1. Name task force to define development tools, including

   software packages
   application literature
   computing power

   to support black box approach to logic design. Task force performs liaison with semi houses.

2. Task force recommends computing power in P30.

3. P30 and Universal Logic Pak is designed and introduced.

4. UniPaks, Gang Modules, card sets designed after P30 introduction.

DEB