

JSP—A Research Signal Processor in Josephson Technology

This paper describes briefly the main architectural and design features of the Josephson Signal Processor (JSP), including its data flow, basic circuit arrangement, and packaging concept. Preliminary partitioning has indicated that, using a 5- μm "single turn logic" technology, the JSP—consisting of about 5000 cells (each with four Josephson junctions) of logic, about 150K bits of nondestructive read out (NDRO) memory, 256K bits of destructive read out (DRO) memory, and 6K bits of read only memory (ROM)—can be packaged in about 20 modules, occupying about 60 cc and consuming about 500 mW of power. The target cycle time for the JSP is 5 ns, with the NDRO and DRO memories having an access/cycle time of about 2.5/4 and 15/30 ns, respectively. Using a 2.5- μm "current injection logic" technology, the JSP—consisting of about 9000 three-junction and 5000 two-junction interferometers—can be packaged in seven modules occupying about 12 cc and will consume about 250 mW of power. The target cycle time for this technology is about 2 ns, with the NDRO and DRO memories having access/cycle times of 0.9/1.4 and 15/30 ns, respectively.

Introduction

The Josephson Signal Processor (JSP) discussed in this paper is a small, special-purpose computer to be built with the Josephson tunneling technology using the Research Signal Processor (RSP) [1] as a precursor. The main purpose of the JSP is to demonstrate the feasibility of using Josephson technology to realize an ultrahigh-speed computer system.

The RSP was conceived and designed by Peled, and it was developed and built (using semiconductor integrated circuits) at this laboratory. Its design has been specially optimized for signal processing applications. While the JSP has the same basic architecture as the RSP, it will differ from the RSP in mainly three respects:

1. It will be provided with a serial-shift capability for large-scale integrated circuit testing and diagnostic purposes;
2. It will have a serialized I/O interface, and
3. It will have an auxiliary store as a supplement to the data store.

This paper describes briefly the RSP/JSP architecture and design, its operation, the incorporation of the serial-shift feature into the JSP, serialization of the I/O interface, the auxiliary store attachment, the packaging concept for the

JSP, the mapping of the RSP circuitry into Josephson technology, the partitioning of the JSP, and finally the estimated size and projected performance of the JSP.

RSP/JSP architecture and design

Figure 1 is a block diagram showing the data flow in the RSP/JSP. The auxiliary store is included only in the JSP.

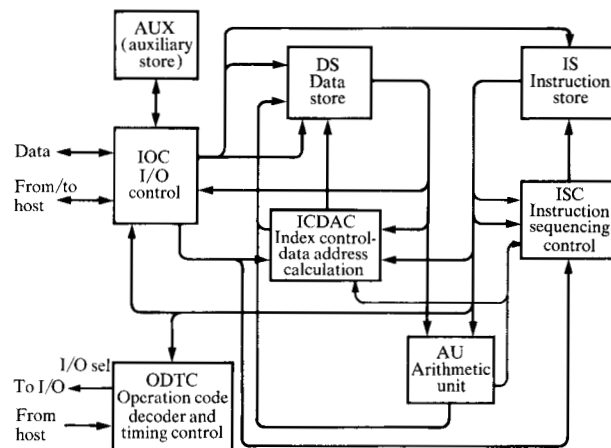


Figure 1 Block diagram of data flow in RSP/JSP.

Copyright 1980 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to republish other excerpts should be obtained from the Editor.

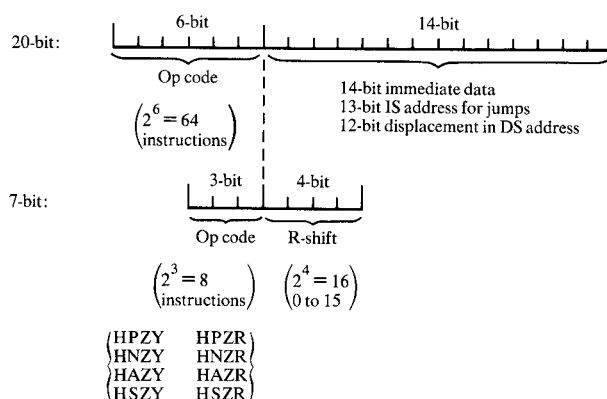


Figure 2 RSP/JSP instruction formats.

CSDC & shift-&-add/sub's for multiply:

To compute: $Y = c_1 z_1 + c_2 z_2 + c_3 z_3 + \dots$

If $c_1 = 011001111001111$

$= 1010100010100001 \Rightarrow 0+, 2-, 4+, 8-, 10+, 15-$

$c_2 = 0011111000111100$

$= 0100001001000100 \Rightarrow 1+, 6-, 9+, 13-$

Program section & fixed-coefficient multiply subroutines:

$0 \rightarrow RY$	$\alpha: [RY] + [RZ]_{-10} \rightarrow RY$
$DS(a_1) \rightarrow RZ$	$[RY] - [RZ]_{-2} \rightarrow RY$
$JMP \alpha$	$[RY] + [RZ]_{-4} \rightarrow RY$
$DS(a_2) \rightarrow RZ \text{ \& } PROCEED$	$[RY] - [RZ]_{-6} \rightarrow RY$
$DS(a_3) \rightarrow RZ \text{ \& } PROCEED$	$[RY] + [RZ]_{-10} \rightarrow RY$
	$[RY] - [RZ]_{-15} \rightarrow RY \text{ \& } RETURN$
	$\beta: [RY] + [RZ]_{-1} \rightarrow RY$
	$[RY] - [RZ]_{-6} \rightarrow RY$
	$[RY] + [RZ]_{-9} \rightarrow RY$
	$[RY] - [RZ]_{-13} \rightarrow RY \text{ \& } RETURN$
	$\gamma: \vdots$

Figure 3 Example illustrating use of CSDC representation of multipliers.

The RSP/JSP is designed for carrying out signal processing computations economically and efficiently. This is achieved by using an architecture intended to facilitate the execution of various Fourier transform computational methods, including the new Winograd-Fourier-Transform (WFT) methods [2]. In general, for most signal processing computations, the multiplications to be carried out can, since the multipliers are normally coefficients known at program-assembly time, be replaced by sequences of consecutive add or subtract operations with concurrent shifts

(thus avoiding the need for an expensive fast multiplier in the hardware) without incurring appreciable loss in performance [1]. Accordingly, the following features have been incorporated in the RSP/JSP design to optimize the execution of signal processing computations:

1. The multiplier coefficients known at program-assembly time are represented in the so-called canonical signed digit code (CSDC) [3], and multiplications with such coefficients are then programmed as sequences of special instructions (subroutines).
2. A shift-and-add-or-subtract subunit in the AU is designed to be operated upon by such special instructions directly.
3. Some special functions (STACK, PROCEED, and RETURN) are provided, which can be invoked either by issuing single instructions or combining them in certain frequently used instructions, to allow fast to-and-fro linkages between main sections of the program and the subroutines for multiplication with known coefficients.
4. An instruction address stack is provided in the instruction sequencing control (ISC) unit to accommodate nesting of such linkages in loops when needed.

There are two formats for the instructions in the RSP/JSP: 20-bit and 7-bit (Fig. 2). In the 20-bit instructions, 6 bits are used to give the operation code, and the remaining 14 bits form an operand which can specify immediate data, an IS address, or a displacement in a DS address. The 7-bit format, an abbreviation of the 20-bit one and used exclusively for the shift-and-add-or-subtract instructions, has a 3-bit operation code and a 4-bit operand, the latter to specify the number of shifts (0 through 15) prior to the add or subtract. Thus, the RSP/JSP instruction set has 64 instructions, including 8 special shift-and-add-or-subtract instructions, and, accordingly, the IS has a $4K \times 20$ -bit and a $1K \times 7$ -bit section.

The data paths in the RSP/JSP are mostly 16 bits wide (DS input and output, computation result, I/O). The ALU, a right and a left shifter, and an intermediate register in the AU are made 20 bits wide in order to reduce accumulation of rounding off errors. The data storage capacities in the JSP will be $4K \times 16$ bits for the DS and $16K \times 16$ bits for the AUX. Thus, the effective widths of addresses are 13 bits, 12 bits, and 14 bits for the IS, the DS, and the AUX, respectively. The JSP will use also about 6K bits of ROM for operation code decoding.

Operation of the RSP/JSP

The RSP/JSP is to be operated as a subsystem under the control of a host processor. The host will load the instruction program into the IS, set up pointer registers for program start and DS accessing, set up routes for data I/O,

and initiate the program for processing in the RSP/JSP. Once initiated, the RSP/JSP will run with its own cycle timing independently of the host.

The RSP/JSP is organized to be operated as a four-phased pipeline. At the beginning of each machine cycle, execution of an instruction is started, which will then take four cycles to be completed. Some instructions may have some idle phases during their execution. The pipeline organization implies, of course, that the four phases can be operating simultaneously, each for one phase of one of four instructions being executed. The allocation of the phases to distinct parts of the RSP/JSP simplifies programming for the pipeline operation; it can be accomplished by a relatively simple set of single operand instructions, and the writing of software tools to aid programming can be considerably facilitated.

Figure 3 illustrates the use of CSDC representation and shift-and-add-or-subtract operation to carry out multiplications with predeterminable multipliers. Figure 4 indicates the use of the STACK, PROCEED, and RETURN instructions to achieve fast subroutine linkages. The instruction address stack provided in the RSP/JSP is four deep, allowing subroutine loops to be nested up to four levels.

The RSP/JSP is intended to be used for signal processing applications, which are predominantly "computation intensive." For processing efficiency, the ratio between the number of machine cycles being used for computation ("number crunching") and the number for I/O should be high (in excess of 50). For simplicity in design, the RSP/JSP does not have the conventional interrupt provisions. Transfer of I/O data to and from the DS in the RSP/JSP takes place by means of cycle stealing. During a stolen cycle, the pipeline is stalled and the DS is allowed to be accessed for an I/O data transfer. The actual timing for the data I/O is under the control of the RSP/JSP. This is to ensure that the acquisition of input data (e.g., by sampling through an A/D converter) or the delivery of output data (e.g., to a D/A converter for display) will be at a constant rate, independently of the cycle stealing.

A load pointer, a compute pointer, and a mask register are provided in the ICDAC unit to allow data to be written into and read from the DS in a "circular" mode (where the addresses will wrap around to keep the accessing within a section of the DS). This gives the RSP/JSP, despite its limited DS capacity, the capability of processing data in a "continuous" mode of operation, which will be useful for applications such as monitoring signal sources.

STACK/PROCEED/RETURN for fast subroutine linkage:

S = STACK: STORE NEXT-INSTR ADDRESS (NIA) BEFORE GOING TO SUBROUTINE.

P = PROCEED: STORE NEXT-INSTR ADDRESS BEFORE RESUMING SUBROUTINE.

R = RETURN: GO TO STACKED ADDRESS AND STORE SUBROUTINE NIA.

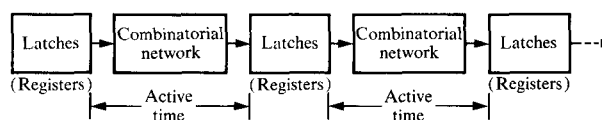
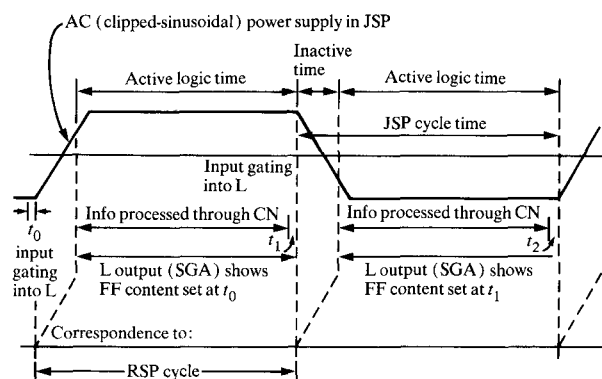
```

i: JMS α      i+1 → STK   α: ...
i+1: ...
      ... R RETURN TO (i+1), & β → STK'
j: ... P      j+1 → STK   β: ...
j+1: ...
      ... R RETURN TO (j+1), & γ → STK'
      γ: ...

```

STACK = 4×13 bits: 4-level loop-nesting.

Figure 4 Use of STACK, PROCEED and RETURN instructions.



AC-supplied JSP logic: active time + inactive time = cycle time

Figure 5 Basic arrangement of logic circuitry in RSP/JSP.

Figure 5 shows the basic arrangement of the logic circuitry of the RSP/JSP. At the beginning of each machine cycle, data stored in registers or latches (Ls) are gated out and fed into combinatorial networks (CNs), whose outputs are subsequently stored into other latches before the end of the cycle. This is prescribed by the use of an alternating current power supply and a latching mode of logic circuit operation in the JSP (see description below and also [4]). In the design and construction of the JSP, with a view to ensuring that the RSP can later be used as an aid in debugging and bringing-up the JSP, register-to-register compatibility with the RSP is being maintained.

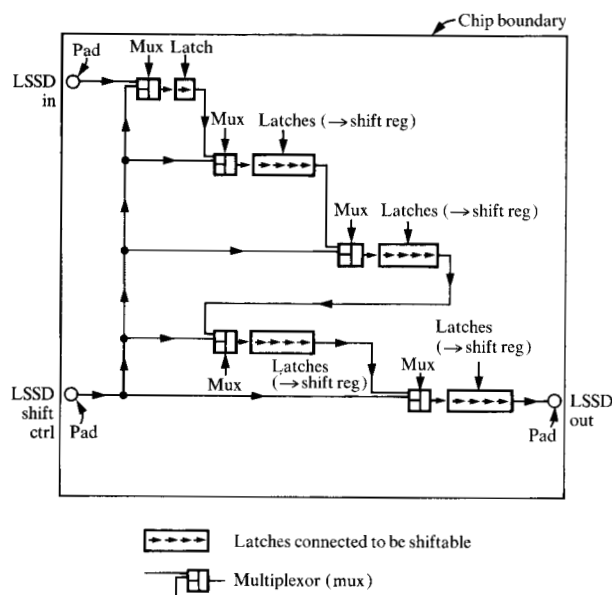


Figure 6 Principle of serial-shift arrangement.

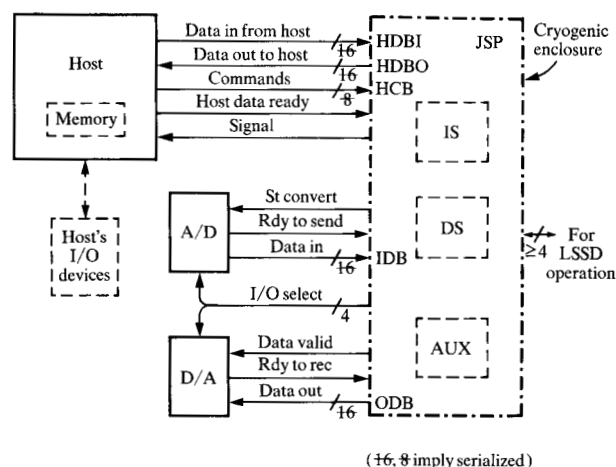


Figure 7 Environment for JSP operation.

Incorporation of serial-shift feature in JSP

The serial-shift feature is an arrangement to provide, under the constraint of I/O limitations, a means for testing and diagnosing LSI circuitry at the chip, module, and/or system levels. It consists essentially (Fig. 6) of connecting all the latches in a chip, in a module, or in the system into one or more shiftable chains and of providing

controls for a test mode of operation which utilizes such chains. In the test mode, the registers will first be set, through serial-shift input, to hold certain contents; then the circuits will be allowed to run through one or more machine cycles, at the end of which the outputs of the combinatorial networks will be stored in registers and finally shifted out to be analyzed.

Obviously, the serial-shift feature requires the use of additional timing and operational controls. The choice of the number of shift-register chains into which the latches in a chip, a module, or the system are to be connected is a matter of tradeoffs among (a) the number of extra I/O ports (chip pads or module pins) used for the serial-shift, (b) the required probable availability and reliability (non-failure rate) of the register elements and serial-shift connections, and (c) the ease of use of the testing arrangement (relative simplicity in generating test patterns and shift controls and the economy in computing time for analyzing test results).

In all known technologies, because of advances in the degree of miniaturization and the density of packaging, LSI circuits have become increasingly inaccessible to direct probing for testing and diagnostics. The need for Josephson technology circuits to be operated within a cryogenic enclosure makes the JSP circuitry inaccessible to external probing in a less conventional manner. The provision and use of the serial-shift feature in the JSP is therefore indispensable.

In the Josephson circuit design, the latch always includes (besides a flip-flop as the storage element and a self-gating AND output [5]) input gating. Thus, the serial-shift paths can readily be incorporated in this input gating design without requiring much additional hardware.

For testing purposes, the serial-shift feature need not be operated at very high speed. In fact the maximum shifting speed is limited to that of the circuits on the room-temperature side of the interface. As will be seen later, the shift chain paths can also be used for I/O interface serialization. The design of the shifting controls in the JSP takes this into account, which is discussed in the next section.

Serialization of I/O interface

Figure 7 is a schematic diagram showing the environment for JSP operation. The JSP, with all its logic circuits and memories (IS, DS, and AUX), will be kept in a cryogenic enclosure. Its I/O interface across the cryogenic/room-temperature boundary will comprise, besides some 20 control signal and power supply lines, data signal lines (72, if furnished for bit-parallel transfers).

In the early days of Josephson technology experimentation, it may have been feared that heat influx through the interface into the cryogenic enclosure could cause unsolvable problems for the refrigeration and packaging. Developments since have indicated that transmission lines can be fabricated on a polyimide substrate, which will admit a heat leak into the enclosure at a rate of less than 1 mW per line. Thus, even if the JSP I/O interface were to require 100 lines to allow fully parallel data transfers, the heat influx would still be tolerable.

On the room-temperature side of the interface, the lines have to be equipped with semiconductor circuits for power level translation, buffering, timing synchronization, etc. Thus, even though a fully parallel I/O interface will not pose any heat influx problems, it is nevertheless desirable to have serialization in the JSP I/O interface for the following reasons:

1. Considerable savings can be made in the semiconductor interface circuits. For the serial-shift feature mentioned earlier, all latches in the JSP will already have been connected to form more than one serial-shift path. The interface serialization can easily share the use of such paths and their shift controls without necessitating a large amount of additional hardware.
2. The serialization will also result in savings in the use of chip pads and module pins.
3. Since the RSP/JSP is intended for applications with high computation-to-I/O ratios, the required average data rate will not need a very fast I/O interface.
4. With the speeds of Josephson technology circuits expected to be very much higher than those of other technologies, it seems expedient not to place excessive demands on the non-Josephson circuits on the room-temperature side of the interface.
5. To prepare for potential future use of the Josephson technology in large ultrahigh-speed computing systems, it will be useful to gather practical experience through the JSP on the design and operation of serialized I/O interfaces. In a future Josephson technology large system, with all its logic, cache and main memories, and backing store realized within the cryogenic enclosure, the I/O interface will be used only for communications with very slow devices (compared with the Josephson technology circuitry), so that it will be even more profitable to have the I/O interface serialized.

The 72 data signal lines that would be needed for bit parallel transfers pertain to five registers: the 16-bit IDB and ODB (input and output data buffers), the HDBI and HDBO (host data buffers for input and output), and the 8-bit HCB (host command buffer). The serialization of

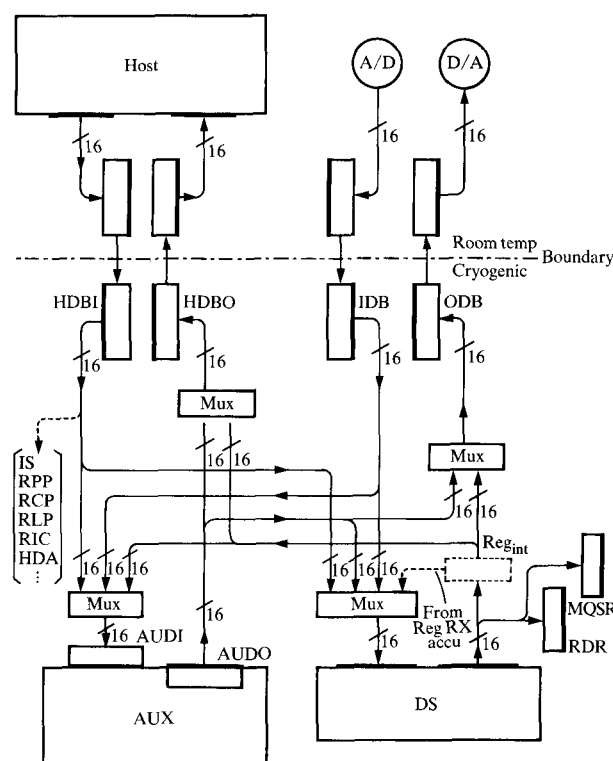


Figure 8 Principle of AUX attachment.

these registers reduces the number of lines to about ten: five signal and five shift-control lines (one for each of the registers).

Auxiliary store and its attachment to JSP

As mentioned earlier, the JSP will include an auxiliary store (AUX) with a capacity of 16K bits $\times$ 16 bits. The AUX will be implemented by using SFQ (single-flux-quantum) DRO memory cells [6], whereas the DS and IS will be NDRO memories using persistent current loop cells [7].

Figure 8 shows the AUX attachment to the JSP. Theoretically, the AUX, which is intended for use as a supplement to the DS, can be looked upon as a backing store and accordingly be treated as an I/O device. In the attachment, however, the AUX (with its data in and data out buffer registers, AUDI and AUDO) is connected, for data transfers with the DS, *not* via the IDB and ODB registers, but rather via multiplexors *directly* to the DS. This design was chosen because the IDB and ODB, being registers at the I/O interface subject to serialized data transfers, are relatively *slow* registers. It will take a time bT to set or unload them by serially shifting data into or out of them, where b is the number of bit positions in each of the regis-

Single Turn Logic (STL):		Current Injection Logic (CIL):	
2-input OR	0.25 cell	2-input OR	1 (3-Jn)
4-input OR	0.75 cell	4-input OR	2 (3-Jn)
2-input AND	0.5 cell	2-(2-in OR)-AND	2 (3-Jn) + 1 (2-Jn)
4-input AND	1.5 cell	4-(2-in OR)-AND	4 (3-Jn) + 3 (2-Jn)
Inverter	0.75 cell	Inverter	2 (3-Jn) + 1 (2-Jn)
Latch	4 cells	Latch	9 (3-Jn) + 5 (2-Jn)
(1 cell = 4 junctions)			

Figure 9 Josephson technology STL and CIL circuit families.

ters and T the period needed for shifting the register chain by one bit. Here, $b = 16$ for both IDB and ODB, and the minimum T is determined by the highest shifting speed obtainable in the non-Josephson circuits on the room-temperature side of the I/O interface. During the time bT , the register IDB or ODB would be busy and would not be usable for a fast bit-parallel transfer between the AUX and the DS. Connecting the AUDI and AUDO via multiplexors directly to the DS will avoid such possible hindrances and allow data transfers between the AUX and the DS to take place at speeds which will then be limited only by the AUX cycle time and not by the IDB setting or ODB unloading time.

For simplicity in the design of the timing controls, the period for the serialized I/O and serial-shift shifting will be chosen to be a simple multiple of the JSP cycle time. A counter with gating will be used to generate the shift control pulses.

Mapping of RSP circuitry into Josephson technology

Implementation of the logic circuits of the RSP in the Josephson technology has been studied using the 5- μm STL (single turn logic) and later the 2.5- μm CIL (current injection logic) circuit families, as described in [4] and [8] and listed in Fig. 9. This is done by mapping out in detail (down to the OR and AND gate level) the Josephson technology circuits needed to perform the same logic functions as done in the RSP and then adding the functions needed for the serial-shift capability, the I/O interface serialization, and the AUX attachment. Certain differences in the way the functions are implemented naturally arise, prescribed by some differences in circuit operation between the Josephson and the semiconductor technologies.

Josephson technology circuits, being configured to be used in the so-called "latching" mode of operation, will be driven by an alternating current power supply, which has a near-trapezoidal (clipped sinusoidal) current wave-

form with alternating polarities (see [9]). In the time during which the power supply reverses its polarity, all logic circuits (except the flip-flops which are built with persistent current loops) will be "reset," in that the interferometer junctions in them will be allowed to return to the superconducting state. This latching mode of operation has the following implications:

1. The machine cycle time (the duration of the trapezoid) for the JSP will in effect be split into two portions: an active logic time (the flat top part of the trapezoid, plus a later fraction of the rising edge) and an inactive time (the falling edge, plus an early fraction of the rising edge). The present design aims to use a power supply with a duty cycle (ratio of the duration of the trapezoid's flat top to that of its base) of approximately 80%.
2. Flip-flops have to be used to store data at least over the duration of the inactive time, in order to prevent any information in the system from becoming lost.
3. Built with latching logic circuits, the combinatorial networks (CNs) must, in order to get their results safely stored, deliver them to the flip-flops in the latches (past the latch's input gating) *before* the end of the active logic time.
4. Since the interferometer junctions in all logic circuits (including those used for the input gating in the latches) can be reset *only once* in every cycle, all flip-flops can change their state *not more than once* in each cycle.
5. Since flip-flops can be made to change their state *any time* (including early) during the active logic time, each latch circuit [10] must be equipped with output gating (self-gating AND), in order to synchronize the outputs with the ac power supply cycle and to ensure that they (both the true and the complement) remain stable over the active logic time, independently of whether its flip-flop changes state during that time. This means in effect that, seen at its output, a latch can show a change in its contents *at most once* each cycle, and this only at the *beginning* of the active logic time.
6. In the Josephson CIL circuit design, the inverter consists essentially of a negation device and a summation gate, connected together in series, where the former is controlled by the signal to be inverted and the latter sums the former's output and a strobe signal input. Because of the latching mode of circuit operation, the strobe signal should always arrive *after* the data signal input. This means that each inverter circuit may individually need an additional timing signal for its proper operation.

Due to the special properties of Josephson circuit operation, the differences in the ways of implementing the RSP and the JSP will mainly be the following:

1. Josephson logic gates are current-controlled devices, which are switched by currents through their control windings. A number of gates can be controlled by a common current signal flowing through their control windings joined in series. Thus, in contrast to most semiconductor circuits (which are voltage-controlled and can be driven in parallel), the Josephson logic circuits use the *serial* mode of input control. This constitutes a major alteration in the mapping from the RSP to the JSP circuitry. In cases where a large number of gates are to be driven by a common signal, care has to be taken to ensure that the propagation delays in the serial input lie within tolerable limits, and, if need be, a parallel-serial mode of input control may be used in order to prevent any of the critical paths in the system from being excessively aggravated through delays in the inputs.
2. The Josephson technology circuits, operating at ultrahigh speeds, have to use properly matched transmission lines for their interconnections. Because of this requirement, certain circuit arrangements used in the RSP (such as simple dot ORs, multiplexors using dot ORing of tristate outputs, etc.) cannot be taken over in the JSP. The dot OR is possible also in the Josephson circuit technology, but it is sometimes subject to certain constraints. Therefore, some of the dot OR arrangements in the RSP will be implemented in the JSP by actually using OR gates.
3. Due to circumstances discussed in (5) above, all latches in the JSP must be operated in accordance with the basic arrangement mentioned earlier and shown in Fig. 5. Certain measures taken in the RSP (such as the use of mid-cycle timing to gate out the contents of registers) cannot be adopted in the JSP. Nevertheless, as mentioned earlier, with a view to using the RSP later as an aid in the debugging and bringing-up of the JSP, compatibility between the RSP and JSP with respect to all the important and relevant registers is being maintained in the JSP design.
4. The requirement for an additional timing signal in the inverter operation, as mentioned in (6) above, may introduce problems in the design for timing signal generation. For reliable inverter operation, a safety margin has to be allowed to accommodate temporal tolerances in the timing signal generated. This margin will appear as an added delay in the inverter. In a circuit arrangement where inversions of signals are needed *in cascade*, the cumulative delay given by the sum of the margins may become too large to be acceptable, especially if the circuit arrangement in question constitutes a part of a critical path in the system. In such cases, it will be preferable to use *dual-rail* logic instead of single-rail. By virtue of the fact that the latch circuit design [5] provides dual-rail outputs, only a small por-

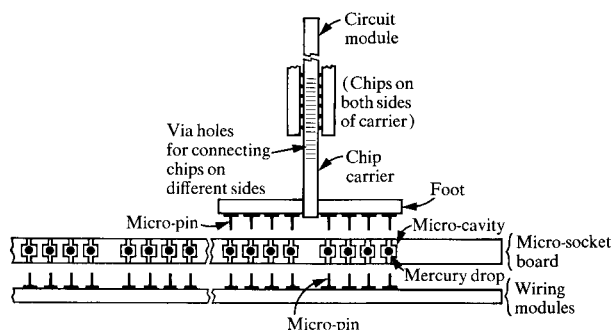


Figure 10 Packaging concept for JSP.

tion of the CNs on the average have to be furnished dual-rail in order to realize all the functions needed in the JSP. It is estimated that implementing the JSP using dual-rail logic without inverters will cost only about 5 to 6% more hardware than using single-rail logic with timed inverters. Therefore, most of the adders and counters in the JSP will be built with dual-rail logic.

5. One interesting characteristic of the Josephson CIL circuit design [8] is that it uses one two-junction and two three-junction interferometers to realize the AND gate; the two-junction interferometer is used for its current summing and threshold detecting capabilities, the three-junction interferometers to standardize the amplitude of the two currents being summed. Since each three-junction interferometer provides a two-input OR without extra cost, the two-input AND gate is actually a two-(two-input OR) AND, and similarly, the four-input AND gate a four-(two-input OR) AND. Thus, in CIL technology, the OR-AND function costs much less to implement than the AND-OR. To take advantage of this relative economy, in the JSP built with 2.5- μ m CIL, multiplexor functions will be realized using OR-AND gates in NIDIC (noninverted-data inverted control) arrangements.

Packaging concept for the JSP

Figure 10 shows the packaging concept for the JSP (see also [10]).

In the case of 5- μ m STL technology, the devices on a chip for implementing logic will be arranged into about 250 cells, with each cell capable of realizing four two-input ORs or two two-input ANDs or with three cells for four inverters or four cells for a latch. An NDRO array chip (for IS and DS) will contain 2K bits, organized as 1K $\times$ 2 bits, and a DRO array chip (for AUX) of 16K bits, organized as 16K $\times$ 1 bit.

In the case of 2.5- μ m CIL technology, the basic devices for implementing logic will be three-junction and two-junction interferometers [11]. About 1536 three-junction and 768 two-junction devices will be placed on a chip and arranged in an array of 64 rows of 24 three-junction and 12 two-junction devices each. For memory, an NDRO array chip (for IS and DS) will contain 4K bits, organized as $1K \times 4$ bits, and a DRO array chip (for AUX) of 16K bits, organized as $16K \times 1$ bit.

The cells or interferometers on a logic chip will be connected together by superconducting metal lines to form the logic circuits as needed [12]. Arranged in double rows along the periphery of the chip, 228 pads will be available for connections to parts external to the logic chip. For the memory chips, 118 (instead of 228) pads, arranged in a single peripheral row, will be provided. The size of the chips will be about 6.5×6.5 mm².

Up to eight chips in the 5- μ m technology and up to 20 chips (10 on each side) in the 2.5- μ m technology will be mounted on a carrier by solder-joining their pads to counterpart pads on the carrier. The sizes of the carriers will be about 30 mm $\times$ 18 mm and 40 mm $\times$ 18 mm, respectively. Via holes through the carrier will be provided to make connections between its two sides. The pads on the carrier for chips will be connected by superconducting metal lines to fillets along an edge of the carrier. The fillets, in turn, will be solder-joined to counterpart fillets on a pedestal foot about 4 mm wide and 30 or 40 mm long; a multitude of micro-pins will be affixed to the foot. Thus, the foot and the carrier (with the chips on it) will form a module.

A board, about 0.4 mm thick and provided with an array of micro-cavities each filled with a mercury drop, will receive the micro-pins from logic and memory modules into one of its sides. Inserted into the other side will be micro-pins from wiring modules furnished with superconducting metal lines to connect the micro-pins with one another. The micro-pins from the logic and memory modules on one side of the board and those from the wiring modules on the other will protrude into the micro-cavities and make contact with one another through the mercury drops. Thus, all the logic and memory modules will be electrically connected together through the board and the wiring modules. Retention hardware will be provided, which will hold, after their assembly, the parts of the package together at room temperature. When the assembled package is placed in the cryogenic enclosure, the mercury drops will become solid and, in addition to providing electrical contacts, will also hold the modules and the board together mechanically. Thus, the packaging method will have the advantage of room-temperature

"pluggability," and the use of pluggable wiring modules will provide the flexibility for accommodating engineering changes in the wiring if needed later.

Partitioning of JSP

Based on the Josephson technology circuitry mapped out, preliminary studies for the possible partitioning of the JSP have been made.

In the case of 5- μ m STL technology, 24 logic chips (with 250 cells per chip and usage of about 80%), 76 (1K- $\times$ 2-bit) NDRO array chips, 16 (16K- $\times$ 1-bit) DRO array chips, 12 memory support chips, and 2 ROM chips—a total of 130 chips—will be needed. These can be packaged into 20 modules: 6 logic/control modules, 6 IS and 4 DS NDRO memory modules, 2 AUX DRO memory modules, 1 ROM module, and 1 power-distribution module.

In the case of 2.5- μ m CIL technology, about 8 logic chips (with about 1500 three-junction and 750 two-junction devices per chip and usage of about 85%), 38 (1K- $\times$ 4-bit) NDRO array chips, 16 (16K- $\times$ 1-bit) DRO array chips, 6 memory support chips, and 1 ROM chip—a total of 69 chips—will be needed. These can be packaged into about 4 modules: 2 modules containing the logic and IS, 1 DS module, and 1 AUX module.

Generally, our studies have confirmed that partitioning depends critically on the following factors:

1. The device density (total number of devices available for use) in a packaging unit (chip or module);
2. The intended percentage usage of the devices in the packaging unit, after allowing a margin to ensure wireability and to accommodate possible changes and contingencies;
3. The number of I/O ports (pads or pins) available for use in the packaging unit;
4. The intended percentage usage of the I/O ports for the packaging unit, after allowing a margin to accommodate possible changes and contingencies;
5. The principal approach chosen for the partitioning: whether by functional grouping or by bit group slicing or by a mixture of the two;
6. Packaging objectives, such as choosing to achieve a minimum number of parts or a minimum number of part numbers, etc.; and
7. Packaging constraints, such as the need for choosing a certain partition and placement of parts in order to minimize the lengths of certain critical paths in the system, etc.

In particular, the studies seem to give the following indications:

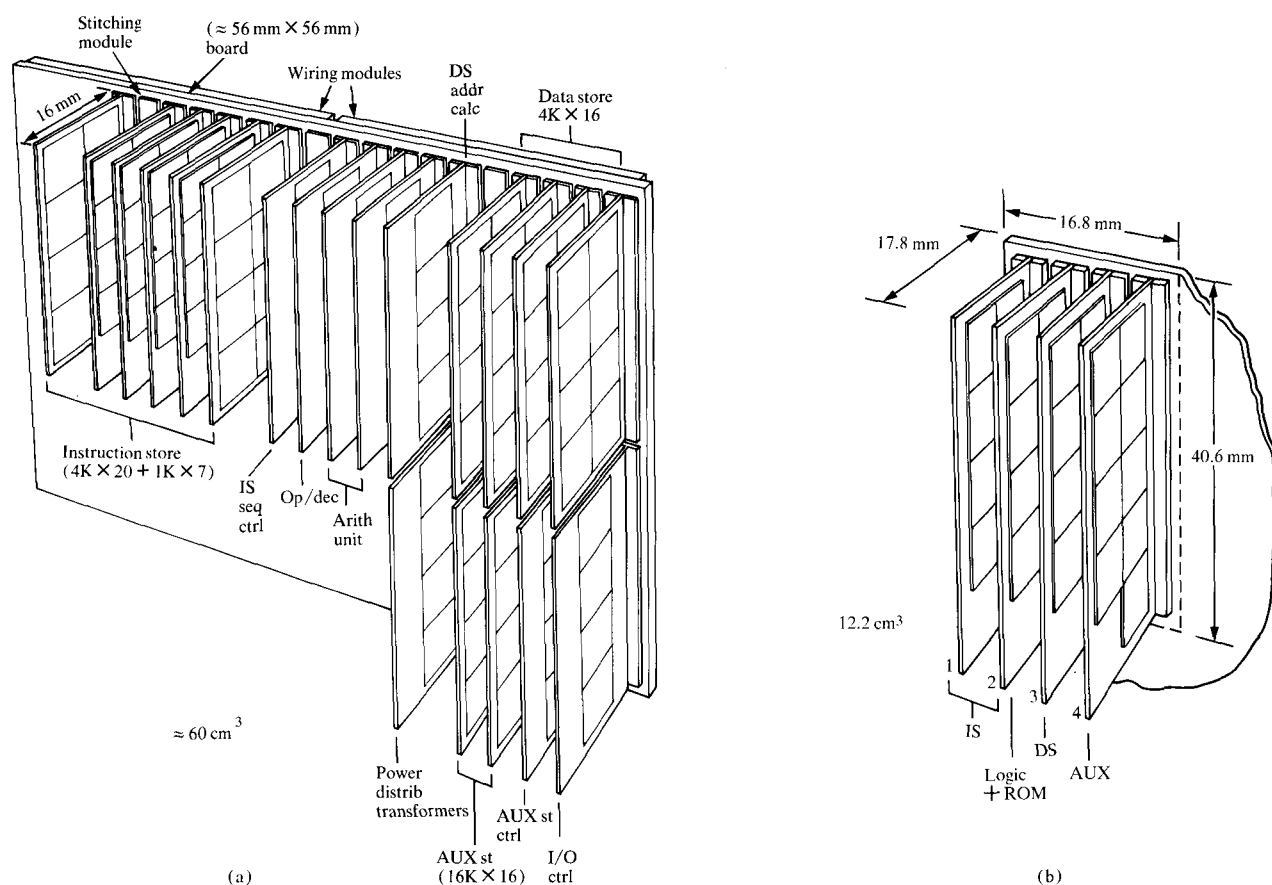


Figure 11 Estimated size of JSP package. (a) 5- μ m STL technology; (b) 2.5- μ m CIL technology.

1. Whenever feasible, the bit-group slicing approach is preferable to the functional grouping approach, since it often helps to reduce the number of I/O paths (pads or pins) needed as well as the number of part numbers.
2. It would be advantageous to try to place, as much as feasible, the circuitry pertaining to a JSP pipeline phase (*i.e.*, entire lengths of a CN's paths, from latch outputs to latch inputs) within the boundary of a packaging unit, since this will be the simplest way to minimize the packaging delays (propagation delays due to interconnections among packaging units).
3. In the case of a control being common to many chips, to avoid excessive delays due to the serial mode of input control, and also to save the use of I/O pads, it would be preferable to distribute the control to the chips in parallel by using a fan-out driver.

Estimated size and projected performance of the JSP

In the 5- μ m STL technology, with the JSP comprising 20 modules, the size of the board will be about 60 mm × 60 mm and the volume of the package will be about 60 cc

[Fig. 11(a)]. Altogether about 5000 cells will be used in the JSP. The target JSP cycle time is about 5 ns. The NDRO memories for the IS and DS will have an access/cycle time of about 2.5/4 ns; the DRO memory for the AUX, about 15/30 ns. Total power consumption will be about 500 mW.

In the 2.5- μ m CIL technology, with the JSP comprising 4 modules, the size of the board will be about 40 mm × 17 mm in size and the volume of the package will be about 12 cc [Fig. 11(b)]. Altogether about 9000 three-junction and 5000 two-junction interferometers will be used in the JSP. The target JSP cycle time is about 2 ns. The NDRO memories for the IS and DS will have an access/cycle time of about 0.9/1.4 ns; the DRO memory for the AUX, about 15/30 ns. Total power consumption will be about 400 mW.

In both cases, an I/O data rate of about 2 megabytes (16 megabits) per second will have to be provided. The shifting speed needed at the serialized I/O interface will there-

fore be about 1 bit every 64 ns. Thus, the shifting control pulses can be easily generated by counting down the JSP clock at the ratio of 12:1 for the 5- μ m STL technology and of 32:1 for the 2.5- μ m CIL technology.

References

1. A. Peled, *Proceedings of the 1976 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 12-15, 1976, pp. 636-639. A. Peled, *IEEE Trans. Acoust., Speech, Signal Processing ASSP-24*, 76-86 (1976).
2. S. Winograd, *Proceedings of the National Academy of Science of the USA*, April 1976, pp. 1005-1006. H. F. Silverman, *IEEE Trans. Acoust., Speech, and Signal Processing ASSP-25*, 152-165 (1977). S. Winograd, *Proceedings of the 1977 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 9-11, 1977, pp. 366-368. H. F. Silverman, *Proceedings of the 1977 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 9-11, 1977, pp. 369-372. S. Winograd, *Math. Comput.* **32**, 175-199 (1978). A. Peled, *Proceedings, 1978 International Symposium on Circuits and Systems*, May 17-19, 1978, pp. 659-661.
3. M. Lehman, *IRE Trans. Electronic Computers EC-6*, 204-205 (1957). M. Lehman, *Proceedings of the Institute of Electrical Engineers (London)* **105B**, 496-503 (1958). G. W. Reitwiesner, *Binary Arithmetic (Advances in Computers, Vol. 1)*, Academic Press, Inc., 1960, pp. 231-308. H. L. Garner, *Number Systems and Arithmetic (Advances in Computers, Vol. 6)*, Academic Press, Inc., 1965, pp. 131-194.
4. M. Klein, D. J. Herrell, and A. Davidson, 1978 IEEE International Solid-State Circuits Conference, *Digest of Technical Papers*, pp. 62-63 and 266. M. Klein and D. J. Herrell, *IEEE J. Solid-State Circuits SC-13*, 577-583 (1978). D. J. Herrell, P. C. Arnett, and M. Klein, Conference on Future Trends in Superconductive Electronics, March 1978, *AIP Conference Proceedings*, No. 44, 470-478 (1978).
5. A. Davidson, *IEEE J. Solid-State Circuits SC-13*, 583 (1978).
6. H. H. Zappe, *Appl. Phys. Lett.* **25**, 424-426 (1974). P. Guéret, Th. O. Mohr, and P. Wolf, *IEEE Trans. Magnetics MAG-13*, 52-55 (1977). R. F. Broom, P. Guéret, W. Korytka, Th. O. Mohr, A. Moser, A. Oosenbrug, and P. Wolf, 1978 IEEE International Solid-State Circuits Conference, *Digest of Technical Papers*, pp. 60-61 and 266. P. Guéret, A. Moser and P. Wolf, *IBM J. Res. Develop.* **24** (1980, this issue).
7. W. H. Henkels and H. S. Zappe, *IEEE J. Solid-State Circuits SC-13*, 591-600 (1978). S. M. Faris and A. Davidson, *IEEE Trans. Magnetics MAG-15*, 416-419 (1979). W. H. Henkels, U.S. Patent No. 4,130,893, 1978. W. H. Henkels, *J. Appl. Phys.* **50** (Dec. 1979). W. H. Henkels and J. H. Greiner, *IEEE J. Solid-State Circuits SC-14*, 794 (1979). S. M. Faris, *IEEE J. Solid-State Circuits SC-14*, 699 (1979). S. M. Faris, W. H. Henkels, E. A. Valsamakis, and H. H. Zappe, *IBM J. Res. Develop.* **24** (1980, this issue).
8. T. R. Gheewala, *Appl. Phys. Lett.* **33**, 781-783 (1978). T. R. Gheewala, *IEEE J. Solid-State Circuits SC-14*, 787 (1979). T. R. Gheewala, *IBM J. Res. Develop.* **24** (1980, this issue).
9. P. C. Arnett and D. J. Herrell, *IEEE Trans. Magnetics MAG-15*, 554-557 (1979).
10. A. Davidson, *IEEE J. Solid-State Circuits SC-13*, 583-590 (1978). A. Davidson and D. J. Herrell, U.S. Patent No. 4,136,290, 1979.
11. Alan V. Brown, *IBM J. Res. Develop.* **24** (1980, this issue). P. Geldermans and C. Y. Ting, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, unpublished results.
12. H. H. Zappe, *Appl. Phys. Lett.* **27**, 432-434 (1975). H. H. Zappe, *IEEE Trans. Magnetics MAG-13*, 41-47 (1977). H. H. Zappe, U.S. Patent No. 4,117,503, 1978. L. M. Gelpert, J. H. Greiner, D. J. Herrell, and S. Klepner, *IEEE Trans. Magnetics MAG-13*, 412-415 (1979).
13. W. Donath, *IBM J. Res. Develop.*, accepted for publication.

Received June 4, 1979; revised September 2, 1979

The author is located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.