# HD81831Y10/Y20(DSP-I)

# Digital Signal Processor Image

The HD81831 (DSP- I ) is a signal chip processor with storedprogram capability designed for digital signal processing including image processing.

The DSP- I can perform multiply and accumulate at a ultra-high speed. Moreover, it achieves high throughput by performing data operation and data transfer simultaneously.

The DSP- I incorporates two-dimensional addressing operation functions and powerful bit processing functions for image a processing, realizing high speed repetitive operations and division. Since its internal memory space consists only of RAM, programs can be changed depending on the application, thus enabling the DSP- I to perform various types of signal processing. Moreover, the DSP- I has a large external data memory space to 16M words, and host MPU can be intefaced to multiple DSP- I s in a multiprocessor system to enhance throughput.

The DSP- I can be widely used for various application requiring high speed and high accuracy operations such as image processing, communications and control systems.

## Hardware Features

- 32-bit high speed fixed point ALU and 16-bit high speed fixed point multiplier incorporated
- Built-in high speed bit processing unit (BPU)
- High speed data processing
- High throughput owing to pipeline operations and horizontal microinstructions
- Two-hierarchical instruction structure allowing data operations and data transfer to occur simultanously
- High speed data transfer achieved with double buffer method using multi-bank large capacity internal data RAM
- Three types of addressing operation units incorporated
- N-bit barrel rotator incorporated
- Wait function for low speed external data memory
- Built-in large capacity memory
  instruction RAM:
    microinstruction: 1024 words × 48 bits
    picoinstruction: 64 words × 16 bits
  data RAM: 512 words × 4 pages × 16 bits
- Large external data memory space of 16M words × 16 bits
- External instruction memory expandable up to 15k words × 48 bits
- 8/16 bit microcomputer interface
- Multiprocessor function
  Control command communication between the host MPU and multiple DSP- I s using processor numbers
  Task communication for cascade connection
- Internal register contents which can be changed forcibly by control commands from the host MPU
- DMA operation with the host MPU or other external devices
- High speed transfer of a large amount of data or program without DMAC(PIO transfer)
- 4 level subroutine nesting
- +5 V single power supply
- Package : 135-pin PGA

## Software Features

- Each operation instruction executed in a single machine cycle.
- Two-hierarchical instruction format: microinstructions and picoinstructions
- Data read, addition/subtraction, multiplication, data write, internal data RAM addressing operations, data transfer to external data memory, external data memory addressing oprations and repeat counter decrementation, all executed by a single program step
- Two-dimensional addressing operation executed by a single program step
- Continuous access operation over a large external data memory space of 16M words
- Three types of repeat functions
  n-step high speed repeat instruction
  loop instruction
  loop due to jump instruction
- Proguram transfer mode (PIO mode) allowing hith speed teansfer between a host MPU and internal data RAM and instruction RAM without DMAC
- Various types of high speed bit processing performed in word units (such as word synthesizing, word matching, run-length measurement)

**Product line-up**

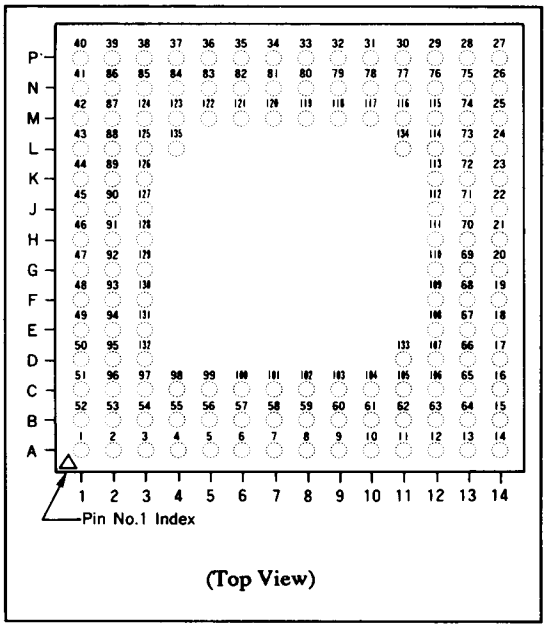
| Product name | Machine cycle | Input clock (CLK1) |
|---|---|---|
| HD81831Y10 | 100ns (10MIPS) | 10 MHz |
| HD81831Y20 | 50ns (20MIPS) | 20 MHz |

## Bench Marks

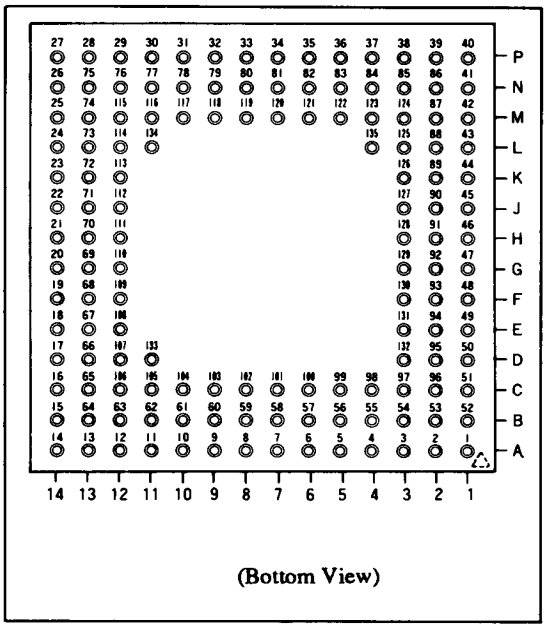
| | | Execution Time | |
|---|---|---|---|
| | | **HD81831Y10** | **HD81831Y20** |
| General Operations (fixed point) | 32 bit ALU operations (ADD, SHIFT, etc.) | 100ns | 50ns |
| | Multiply and accumulate | 100ns | 50ns |
| | 16×16−>32 bit multiply | 100ns | 50ns |
| | Double length multiply (32×32−>64 bits) | 5.6μs | 2.8μs |
| | Double length multiply (32×16−>29 bits) | 0.9μs | 0.45μs |
| | Double length multiply (32×16−>32 bits) | 1.7μs | 0.85μs |
| | Double length multiply (32×30−>32 bits) | 2.5μs | 1.25μs |
| | Divide (32/16 bits) (quotient only) | 2μs | 1μs |
| | Divide (32/16 bits) (both quotient and remainder produced) | 3.7μs | 1.85μs |
| | Square root | 8.3μs | 4.15μs |
| | Trigonometric functions (SIN, COS) | 5.4μs | 2.7μs |
| | Matrix operations (16×16) | 0.62μs | 0.31μs |
| General Operations (floating point) | Multiply (28 bit mantissa field, 16 bit exponent field) | 8.7μs | 4.35μs |
| | Add/subtract (28 bit mantissa field, 16 bit exponent field) | 6.6μs | 3.3μs |
| | Divide (28 bit mantissa field, 16 bit exponent field) | 5.7μs | 2.85μs |
| | Square (28 bit mantissa field, 16 bit exponent field) | 8.8μs | 4.4μs |
| Filter Operations (fixed point) | 256 point complex FFT | 1.30ms | 0.65ms |
| | 512 point complex FFT | 2.90ms | 1.45ms |
| | 1024 point complex FFT | 7.66ms | 3.83ms |
| | Transversal filter (Filter degree; 2) | 7.9μs | 3.95ms |
| | Bi-cut filter (single stage) | 3.0μs | 1.5μs |
| Image Processing Operations | Affine transform (black and white, 2 levels, using external data memory) | 1600ns/PIXEL | 800ns/PIXEL |
| | Spatial filter (3×3 pixels, filter, using external data memory) | 3.0μs/PIXEL | 1.5μs/PIXEL |
| | Histogram conversion (512×512 pixels, using external data memory) | 315ms | 157.5ms |

Note: Refer to the "HD81831 Application Note" document for details on the above benchmarks.

## Pin Arrangement

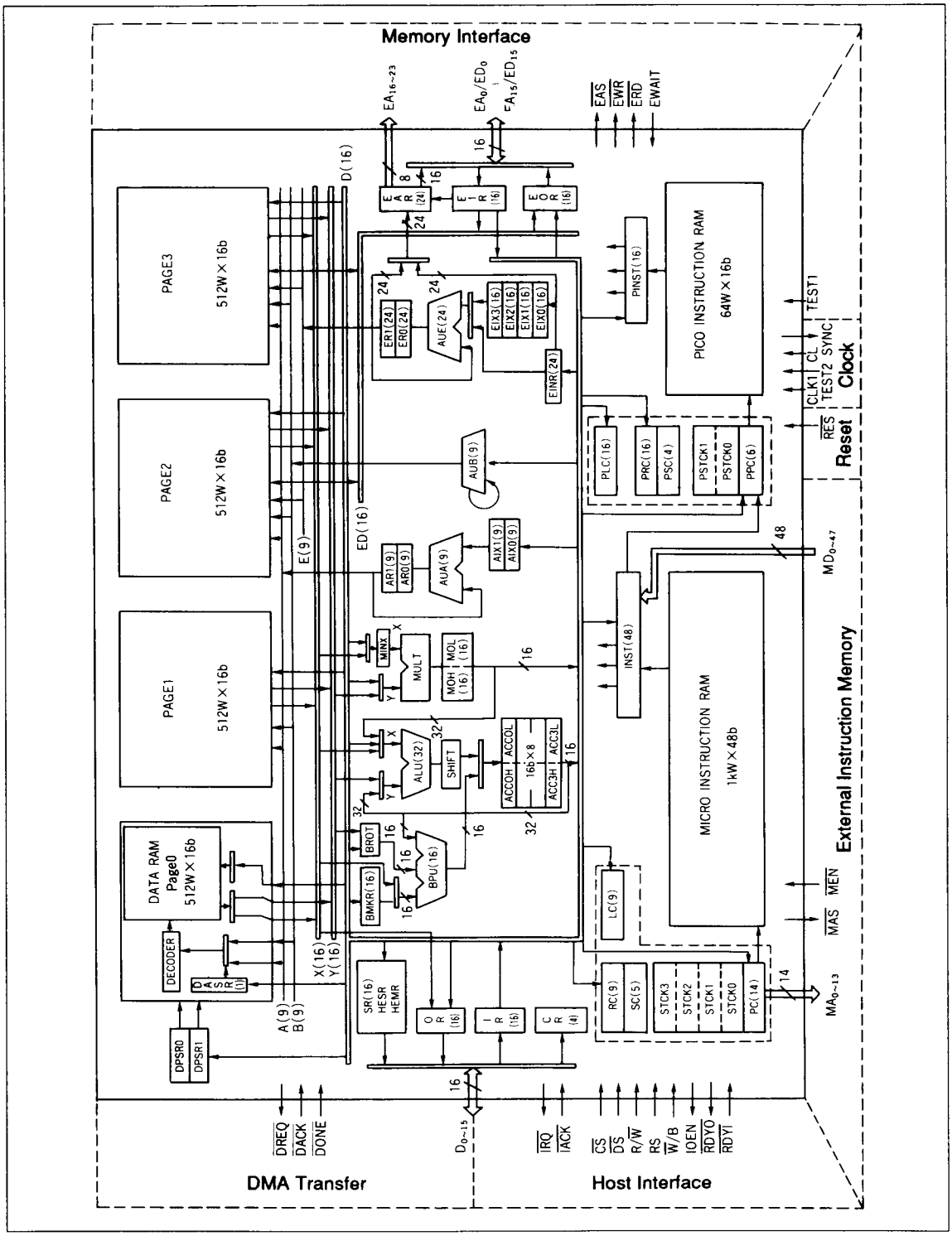



(Top View)



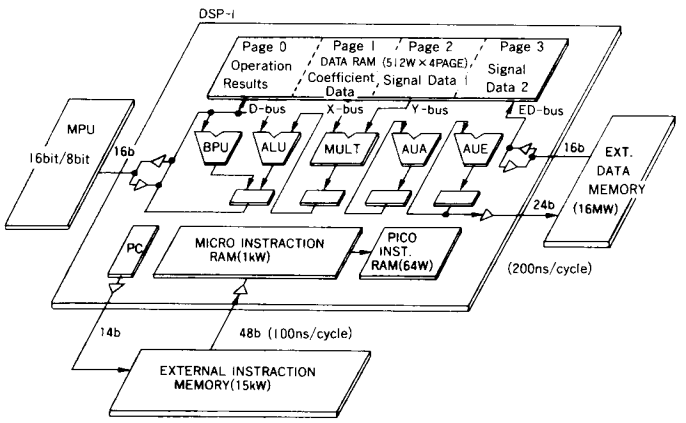

(Bottom View)

## Block Diagram

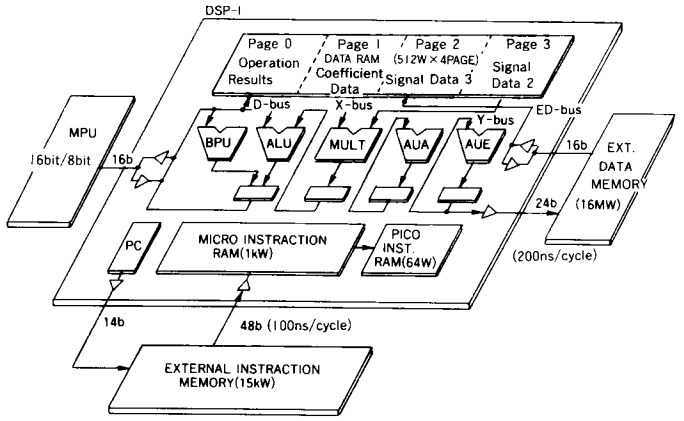


## Architecture
## Data Operation and Data Transfer (Executed in Parallel)

The DSP- I provides microinstructions controlling data operations and picoinstructions controlling data transfer between internal data RAM and external data memory. These two instruction sets can operate independent of each other. In addition, the DSP- I includes 4 pages of internal data RAM, which can be used as a data cache by assigning each page to be controlled by microinstruction or picoinstruction. Therefore, using these functions, the DSP- I can perform data operations in paralled with data transfer between internal data RAM andexternal data memory. This enables high data throughput.




(a) The DSP- I performs data operations using signal data 1. Simultaneously, it transfers signal data 2 required for thenextdata operation from external data memory into page 3.




(b) After completing data operations (a), the DSP- I interchanges page 2 and page 3. It then executes data operations using signal data 2 of page 3. Simultaneously, it transfers signal data 3 from external data memory to page 2 preparatory to the nextdata operation.

*The External Expansion Memory Access cycle shown if for the HD81831Y20. The memory access cycle for the HD81831Y10 is twice.

**Figure 1. Example of Double Buffer Operation**

## Double-Hierarchical Format Instructions

Operations of microinstructions and dicoinstructions are shown in figure 2. A picoinstructions is activated by a microinstruction, after which it operates independent of microinstruction. A picoinstructions is activated by describing the address of the to picoinstructions to be activated in the microinstruction. To synchronize the operations of microinstructions and picoinstructions, microinstruction execution must be halted by the WAIT instruction to wait for the picoinstruction to stop.

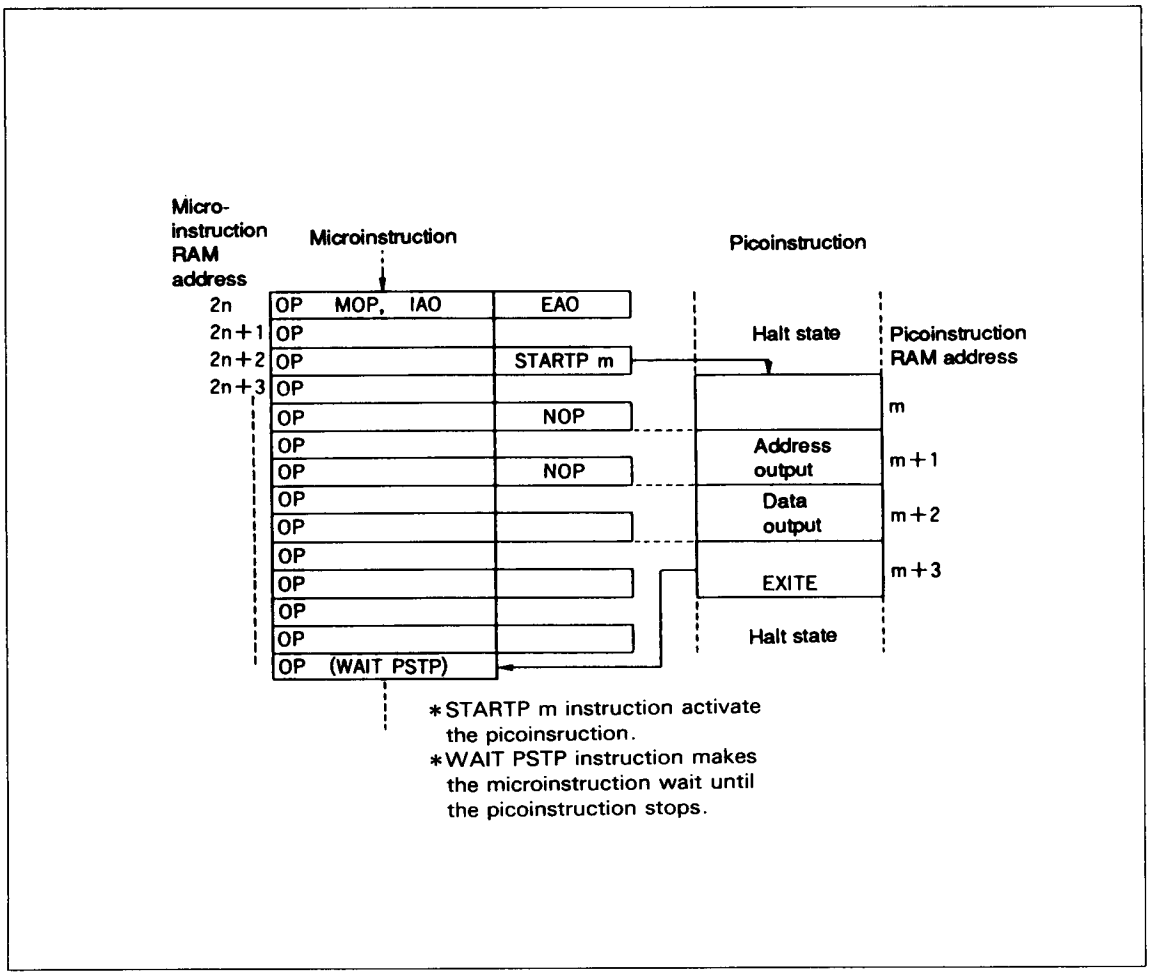



**Figure 2. Microinstructions and Picoinstructions Operation**

## Multiprocessor System

The host MPU communicates with the DSP- I using control commands.
The DSP- I has a 12 bit processor number expressed two-dimensionally (DSP- I identification number). Thus it can be used in a multiprocessor system consisting of up to 64 ×64 DSP- I s (See figure 3). In a multiprocessor system, all DSP- I s are connected to the same system bus. Each DSP- I uses its processor number to determine whether or not the host MPU control commands are given to itself.
Processor number of zero is used for global communication. As figure 3 shows, both X number are Y number are specified when a single DSP- I communicates withe the host MPU, but either X number or Y number is zero when the host MPU communicates with all DSP- I s along a row (X diretion) or column (Y direction). In addition, when both X and Y numbers are zero, all DSP- I s communicate with the host MPU.

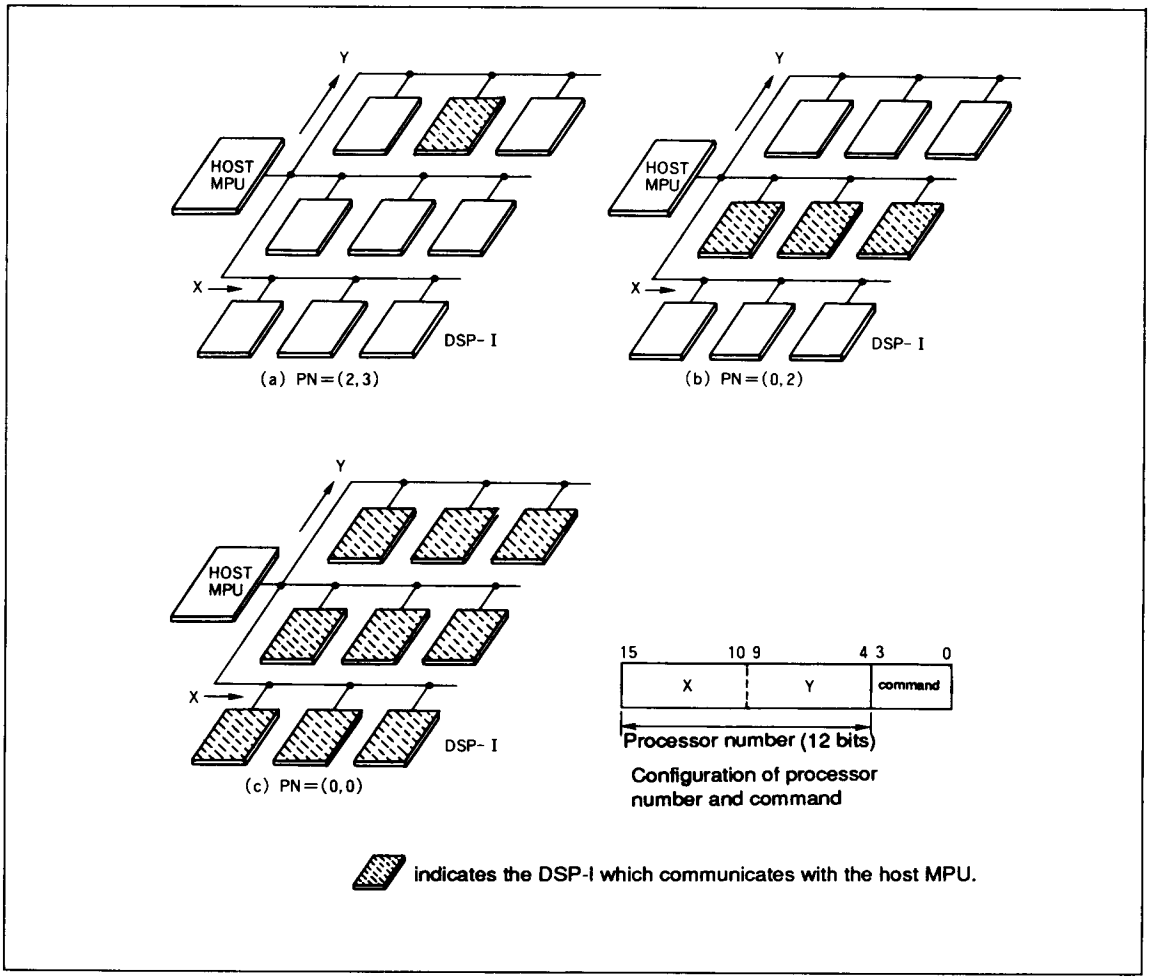



**Figure 3. Example of Multiprocessor System**

### Instruction Execution Cycle Time and Memory Access Time

The DSP- I provides two types of instruction sets: microinstructions and picoinstructions, and incorporates instruction memory for each instruction set. The DSP- I is characterized by ultra-high speed operations, but it can use low-speed external memory. Table 1 summarizes instruction execution cycle tiem memory accesstimes.

**Table 1. Instruction Execution Cycle Time (When basic clock cycle time is 50 ns)**

| Instruction memory | Microinstruction execution | Data operation and internal data RAM addressing execution time | External data memory access time | |
|---|---|---|---|---|
| | | | Microinstruction (EAO field) | Picoinstruction |
| Internal | 50ns | 50ns | 200ns | 200ns |
| External | 100ns | 100ns | 400ns | |

*Values for the HD81830Y10 are twice of those listed in the above table.

### Bit processign unit (BPU)

The DSP- I has a high-performance bit processing unit (BPU) which realizes a high-speed image processig. Figure 4 shows block diagram of the bit processing unit. Table 2 lists examples of processing.

This bit processing unit consists of the following:

### Bit rotator (BROT)

The bit rotator (BROT) otates left 16-bit data (bits 0-15) from Y bus or D bus. It also provides bit rotation and bit reverse in one instruction cycle. The BROT instruction or the rotate register (BRTC, 4 bits) specifies the rotate value as an immediate value. The BRTC is a counter register which is incremented while shifting data. The rotated data is stored in the accumulator or used in an operation with the other operand.

### Mask data generator (MSKG)

The mask data generator (MSKG) is used to generate mask data depending on rotate value N.
These mask data reaizes high-speed performance : synthesis of two word data and bit data substitution.

### Mask data register (BMKR)

This register stores an optional mask data in addition to the mask data generated by the MSKG. The MBKR is set by the MOV instruction.

### Logic operation block

This block performs a logic operation (AND, OR, XOR) between BROT data and accumulator data (or the result and AND of mask data and ACC), which realizes data synthesis.

### Priority Encoder (PE)

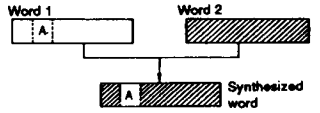
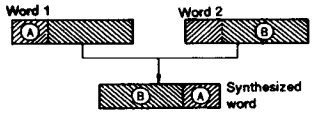
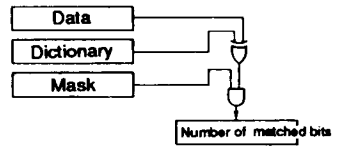
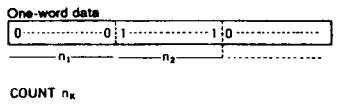
This block performs priority encoding data via BROT, priority encoding selects the first bit location to be set to 1 from the MSB side. The PE realizes data normalization and run length measurement.

| PE input data | PE output data |
|---|---|
| 0000000000000000 | $0000 |
| 1000000000000000 | $0001 |
| 0100000000000000 | $0002 |
| ⋮ | ⋮ |
| 0000000000000001 | $0010 |

### Bit counter

This block is used to count the number of bits set to 1 in word data via BROT, which realizes pattern matching between two word data.

**Table 2. Examples of Bit Processings**

| Processing | Number of Steps | | Description |
|---|---|---|---|
| | DSP-I | HD61810 | |
| Bit substitution | 1 | 3 |   |
| Two-word synthesis | 2 | 17 |   |
| Pattern matching with masking | 2 | 51 |   |
| Run length measurement | 7 | 64 |   |

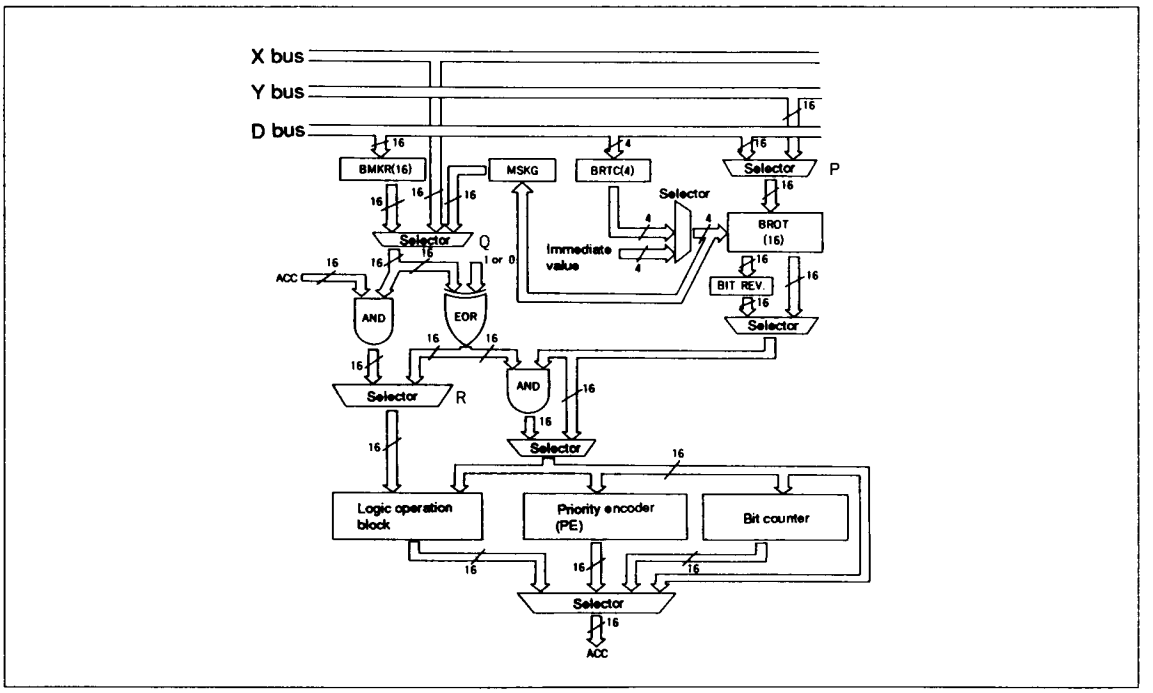



**Figure 4. BPU (Bit Processing Unit) Configuration**