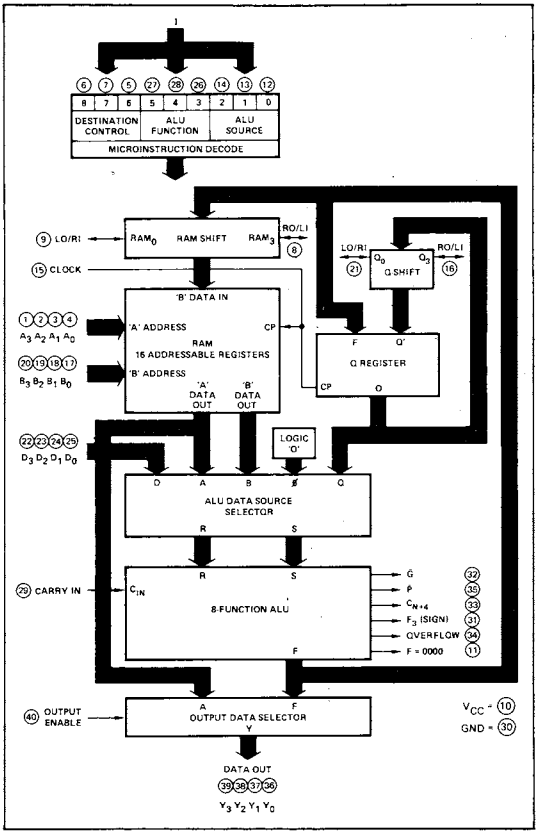
Am2901A

## BLOCK DIAGRAM




## NEW ARITHMETIC CIRCUITS COMING

Am2903 Four-Bit Slice with Expandable Register File
Am2904 Status and Shift Control

## INSTRUCTIONS

### Destination

| $I_{876}$ | Mnemonic | Load | Y= |
|---|---|---|---|
| 0 | LQ | F→Q | F |
| 1 | NOP | – | F |
| 2 | LRA | F→B | A |
| 3 | LRF | F→B | F |
| 4 | LRQD | F/2→B, Q/2→Q | F |
| 5 | LRD | F/2→B | F |
| 6 | LRQU | 2F→B, 2Q→Q | F |
| 7 | LRU | 2F→B | F |

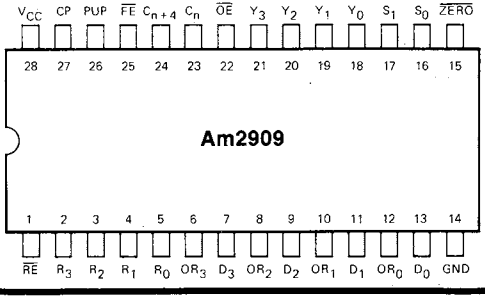
### Function

| $I_{543}$ | Mnemonic | Function |
|---|---|---|
| 0 | ADD | R + S |
| 1 | SUBR | R – S |
| 2 | SUBS | S – R |
| 3 | OR | R ∨ S |
| 4 | AND | R ∧ S |
| 5 | NOTRS | $\overline{R}$ ∧ S |
| 6 | EXOR | R ∀ S |
| 7 | EXNOR | $\overline{R \forall S}$ |

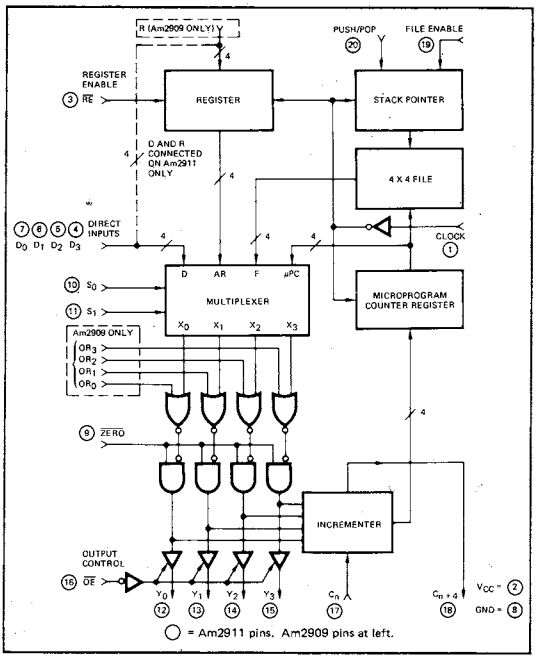
### Source

| $I_{210}$ | Mnemonic | Source R | Source S |
|---|---|---|---|
| 0 | AQ | A | Q |
| 1 | AB | A | B |
| 2 | ZQ | 0 | Q |
| 3 | ZB | 0 | B |
| 4 | ZA | 0 | A |
| 5 | DA | D | A |
| 6 | DQ | D | Q |
| 7 | DZ | D | 0 |

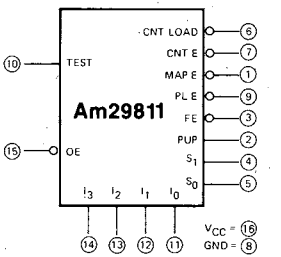
## Am2909 PIN CONNECTIONS

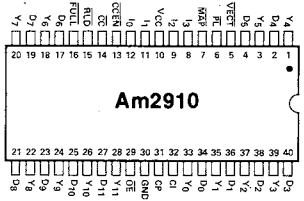


## BLOCK DIAGRAM




## PIN CONNECTIONS




## INSTRUCTIONS

### Address Selection

| OCTAL | $S_1$ | $S_0$ | SOURCE FOR Y OUTPUTS | SYMBOL |
|---|---|---|---|---|
| 0 | L | L | Microprogram Counter | μPC |
| 1 | L | H | Register | REG |
| 2 | H | L | Push-Pop Stack | STK0 |
| 3 | H | H | Direct Inputs | $D_i$ |

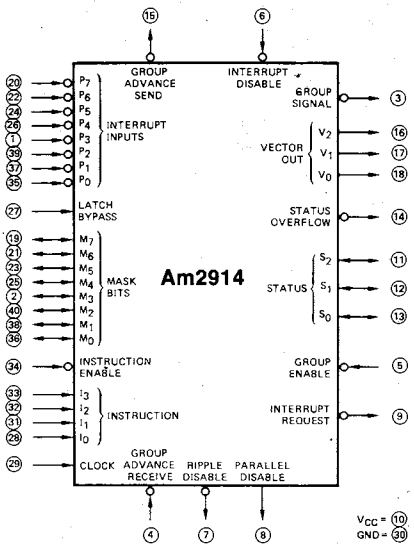
### Synchronous Stack Control

| $\overline{FE}$ | PUP | PUSH-POP STACK CHANGE |
|---|---|---|
| H | X | No Change |
| L | H | Increment stack pointer, then push current PC onto STK0 |
| L | L | Pop stack (decrement stack pointer) |

## INSTRUCTIONS

| MNEMONIC | $I_3$ $I_2$ $I_1$ $I_0$ | INSTRUCTION | |
|---|---|---|---|
| JZ | L L L L | Jump to Address Zero | |
| CJS | L L L H | Conditional Jump-to-Subroutine with Jump Address in Pipeline Register. | |
| JMAP | L L H L | Jump to Address at Mapping PROM Output. | |
| CJP | L L H H | Conditional Jump to Address in Pipeline Register | |
| PUSH | L H L L | Push Stack and Conditionally Load Counter | |
| JSRP | L H L H | Jump-to-Subroutine with Starting Address Conditionally Selected from Am2911 R-Register or Pipeline Register. | |
| CJV | L H H L | Conditional Jump to Vector Address. | |
| JRP | L H H H | Jump to Address Conditionally Selected from Am2911 R-Register or Pipeline Register. | |
| RFCT | H L L L | Repeat Loop if Counter is not Equal to Zero. | |
| RPCT | H L L H | Repeat Pipeline Address if Counter is not Equal to Zero. | |
| CRTN | H L H L | Conditional Return-from-Subroutine. | |
| CJPP | H L H H | Conditional Jump to Pipeline Address and Pop Stack. | |
| LDCT | H H L L | Load Counter and Continue. | |
| LOOP | H H L H | Test End of Loop. | |
| CONT | H H H L | Continue to Next Address. | |
| JP | H H H H | Jump to Pipeline Register Address. | 29811 Only |
| TWB | H H H H | Three Way Branch — To PC if Pass Test, to Loop if Fail Test and CNT ≠ 0, to Pipeline if Fail Test and CNT = 0 | 2910 Only |

## LOGIC SYMBOL




## INSTRUCTIONS

| Decimal $I_3I_2I_1I_0$ | Mnemonic | Instruction |
|---|---|---|
| 0 | MCLR | Master clear |
| 1 | CLRIN | Clear interrupt requests |
| 2 | CLRMB | Clear interrupts from M-bus |
| 3 | CLRMR | Clear interrupts from mask register |
| 4 | CLRVC | Clear interrupt last read |
| 5 | RDVC | Read vector |
| 6 | RDSTA | Read Status |
| 7 | RDM | Read mask |
| 8 | SETM | Set all mask |
| 9 | LDSTA | Load status |
| 10 | BCLRM | Bit clear mask |
| 11 | BSETM | Bit set mask |
| 12 | CLRM | Clear all mask |
| 13 | DISIN | Disable requests |
| 14 | LDM | Load mask |
| 15 | ENIN | Enable requests |

# Am2900

# BIPOLAR MICROPROCESSOR FAMILY

# MICRO PROGRAMMING CARD

ADVANCED MICRO DEVICES

AMDASM

## COMMAND SUMMARY

Δ = required space { } = optional

| DEFINITION PHASE | |
|---|---|
| TITLEΔ | max 60 characters |
| WORDΔn | n ≤ 128 |
| EQUΔ | name:EQUΔconstant/expression |
| SUBΔ | name:SUBΔfield,..10 fields max |
| DEFΔ | name:DEFΔfield,..30 fields max |
| NOLIST | do not print following stmts |
| LIST | print following statements |
| END | end of definition source file |
| **ASSEMBLY PHASE** | |
| TITLEΔ | maximum 60 characters |
| EQUΔ | name:EQU constant/expression |
| NOLIST | do not print following stmts |
| LIST | print following statements |
| f.n.Δ | format name ΔVFS,..(from DEF) |
| FFΔ | free format FFΔfield,..max 30 |
| SPACEΔn | spaces n blank lines |
| EJECT | ejects page |
| ORGΔn | resets program counter (forward) |
| RESΔn | reserves n words of code |
| *ALIGNΔn | sets PC to next even multiple of n |
| **DATAΔn | constant, right just. in word |
| **DUPΔn | duplicates next word n times |
| LABEL: | precedes f.n. or FF, value=PC |
| *LABEL:: | entry point for mapping PROM |
| ; | comment statement |

*not available on AMDASM/TS
**not available on AMDASM/80

names = 8 characters, no blanks
char 1 = A-Z, or . char 2-8 = A-Z,1-9, or .

## FIELD & OPERATOR INFORMATION

| CONSTANTS, EXPRESSIONS, CONSTANT FIELDS |
|---|
| {n} des digits {mod} |
| **VARIABLE FIELDS** |
| n V {attr} {des} {digits} {mod} (digits are default value) |
| n V {attr} X (defaults to X) |
| max n = 16 (AMDASM/80), = 32 (AMDASM/TS) |
| **DON'T CARE FIELDS** |
| n V {attr} X max n = word size |

**MODIFIERS (mod) and ATTRIBUTES (attr)**

| | |
|---|---|
| * | inversion |
| – | negation |
| % | right justify or field has expression |
| : | truncation |
| $ | paging (relative addressing) ATTRIBUTE only, sets % and : |

**EXPRESSION OPERATORS**

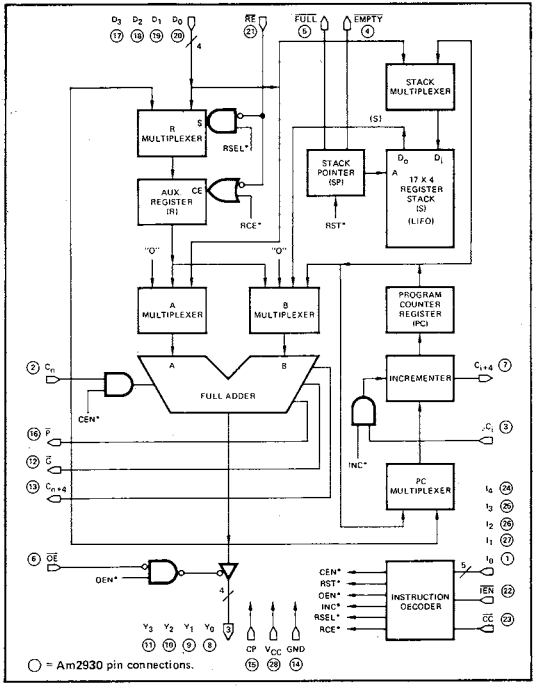
| | | |
|---|---|---|
| + | add | evaluated left to right |
| – | subtract | |
| * | multiply | |
| / | divide | |

| DESIGNATORS (des) | VARIABLE FIELD SUBSTITUTE (VFS) |
|---|---|
| B# binary | label |
| D# decimal | label$ |
| Q# octal | expression |
| H# hexadecimal | digits |
| | des digits {mod} |
| | constant name |

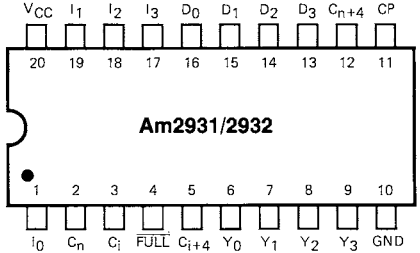
{ } = optional
des = designator
attr = attribute
mod = modifier
digits = numbers

Am2930/31/32

## BLOCK DIAGRAM




## Am2931/32 PIN CONNECTIONS




## INSTRUCTIONS

| Am2930 $I_4I_3I_2I_1I_0$ | $\overline{CC}$ | $\overline{IEN}$ | Am2931 $I_3I_2I_1I_0$ | Am2932 $I_3I_2I_1I_0$ | Mnemonic | Instruction | $Y_{0-3}$ | PC |
|---|---|---|---|---|---|---|---|---|
| all | X | H | – | – | | Disable | Notes 1, 3 | NC |
| 0 | X | L | 0 | 0 | PRST | Reset | 0 | PC + $C_i$ |
| 1 | X | L | 1 | 4 | FPC | Fetch PC | PC | |
| 2 | X | L | – | 8 | FR | Fetch R | R | |
| 3 | X | L | 2 | – | FD | Fetch D | D | |
| 4 | X | L | 4 | – | FRD | Fetch R + D | R+D+$C_n$ | |
| 5 | X | L | 3 | – | FPD | Fetch PC+D | PC+D+$C_n$ | |
| 6 | X | L | – | 9 | FPR | Fetch PC+R | PC+R+$C_n$ | |
| 7 | X | L | – | – | FSD | Fetch S+D | S+D+$C_n$ | |
| 8 | X | L | – | 10 | FPLR | Fetch PC→R | PC | |
| 9 | X | L | 5 | – | FRDR | Fetch R+D→R | R+D+$C_n$ | |
| 10 | X | L | 12 | 15 | PLDR | Load R | PC | |
| 11 | X | L | – | 6 | PSHP | Push PC | PC | |
| 12 | X | L | – | 2 | PSHD | Push D | PC | |
| 13 | X | L | 13 | 3 | POPS | Pop S | S | |
| 14 | X | L | – | – | POPP | Pop PC | PC | |
| 15 | X | L | – | – | PHLD | Hold | PC (Note 3) | NC |
| 16–31 | H | L | – | – | – | Fail Condition (do FPC) | PC | Y + $C_i$ → PC |
| 16 | L | L | – | 11 | JMPR | Jump R | R | |
| 17 | L | L | 6 | 5 | JMPD | Jump D | D | |
| 18 | L | L | – | – | JMPZ | Jump zero | 0 | |
| 19 | L | L | 8 | – | JPRD | Jump R+D | R+D+$C_n$ | |
| 20 | L | L | 7 | – | JPPD | Jump PC+D | PC+D+$C_n$ | |
| 21 | L | L | – | 12 | JPPR | Jump PC+R | PC+R+$C_n$ | |
| 22 | L | L | – | 13 | JSBR | JSB R | R | |
| 23 | L | L | 9 | – | JSBD | JSB D | D | |
| 24 | L | L | – | – | JSBZ | JSB zero | 0 | |
| 25 | L | L | 11 | – | JSRD | JSB R+D | R+D+$C_n$ | |
| 26 | L | L | 10 | – | JSPD | JSB PC+D | PC+D+$C_n$ | |
| 27 | L | L | – | 14 | JSPR | JSB PC+R | PC+R+$C_n$ | |
| 28 | L | L | 14 | 7 | RTS | Return S | S | |
| 29 | L | L | – | – | RTSD | Return S+D | S+D+$C_n$ | |
| 30 | L | L | – | – | CHLD | Hold | PC (Note 3) | NC |
| 31 | L | L | 15 | 1 | PSUS | Suspend | Z (Note 2, 3) | NC |

Notes:
1. When $\overline{IEN}$ is HIGH, the Y outputs contain the same data as when $\overline{IEN}$ is LOW, as determined by CC and $I_{0-4}$.
2. Z = High impedance state (OFF).
3. Clock is disabled internally.

## NEW CIRCUITS COMING

Am2934 Program Control Unit
Am2940 DMA Address and Word Count Generator