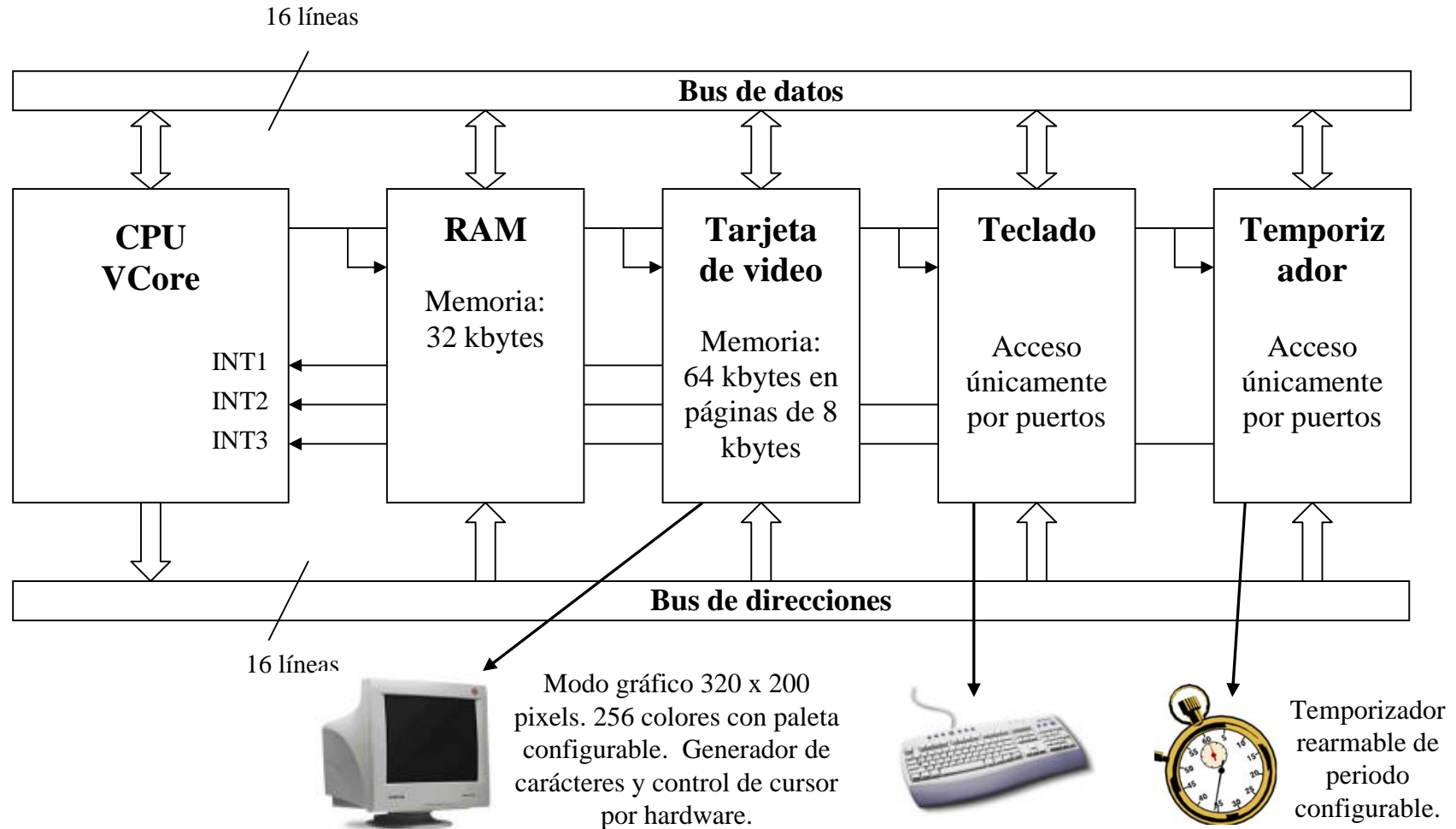


VCore

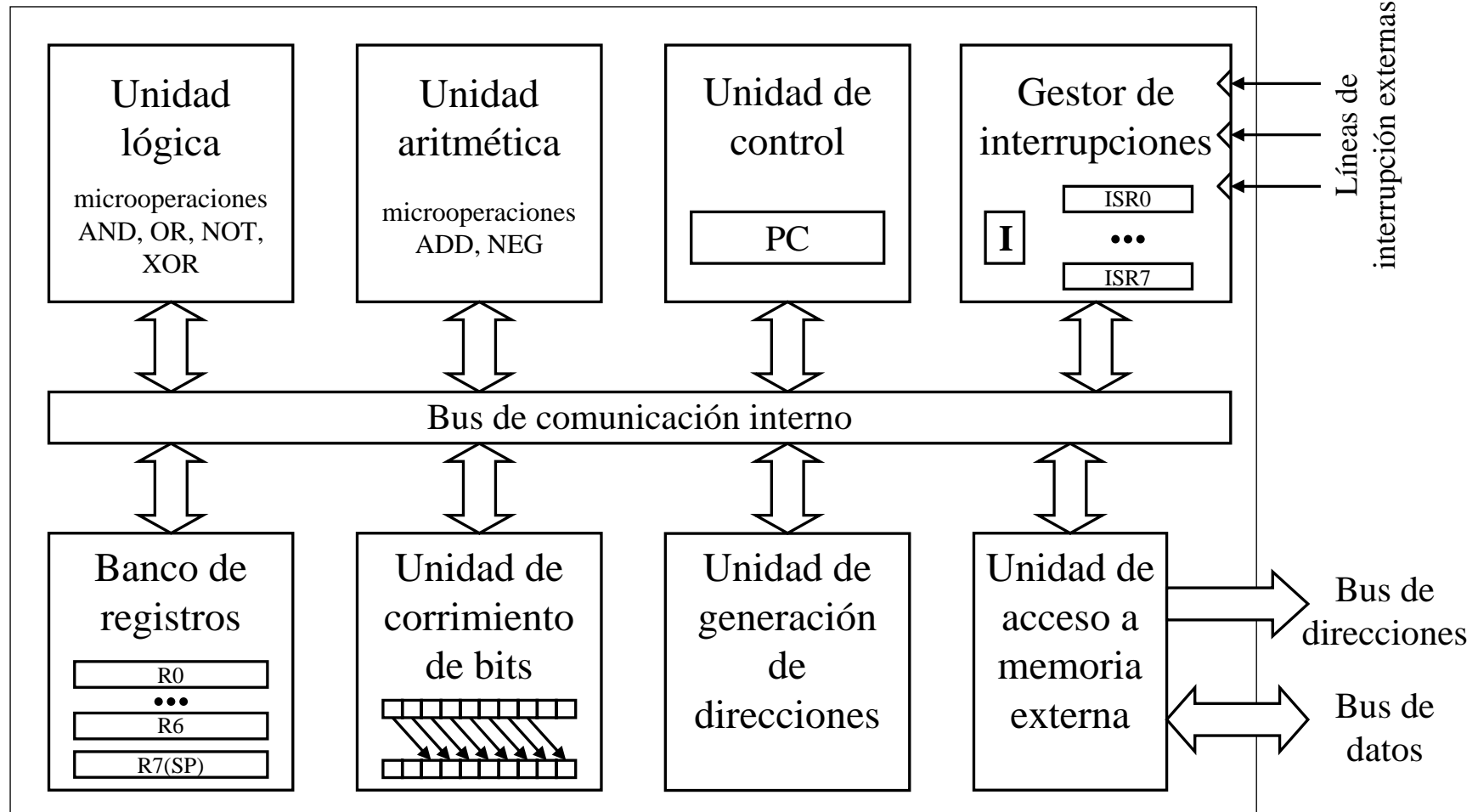
Un procesador de 16 bits de propósito didáctico.

- Modelo estructural de la computadora y del procesador.
- Modelo funcional. Direccionamiento. Operandos y registros.
- Repertorio de instrucciones del procesador.
- Indicadores de la ALU. Ejemplos de procesamiento.
- Notación en ensamblador para VCore.
- Ejemplos de programas: suma en triple precisión.

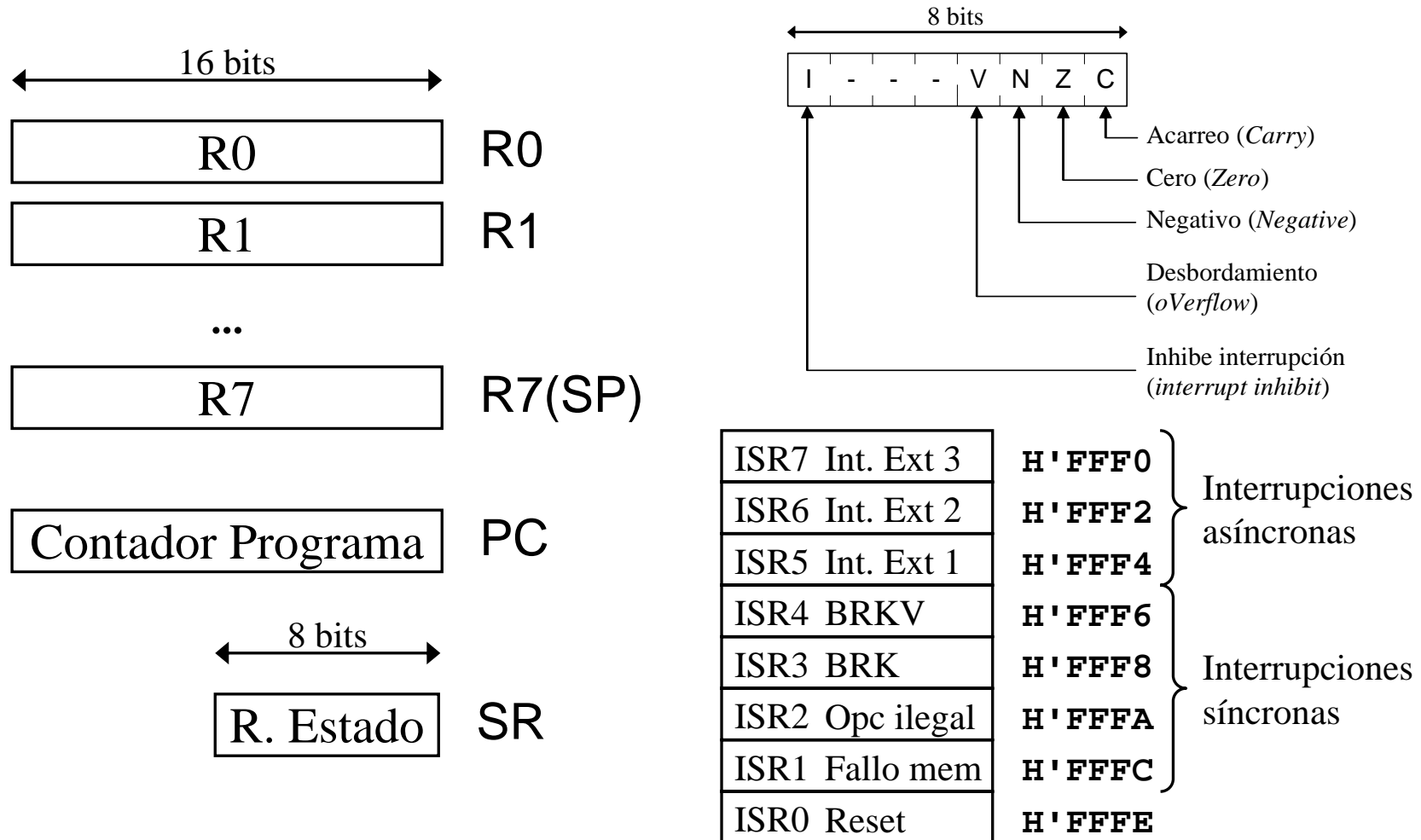
Modelo estructural de la computadora



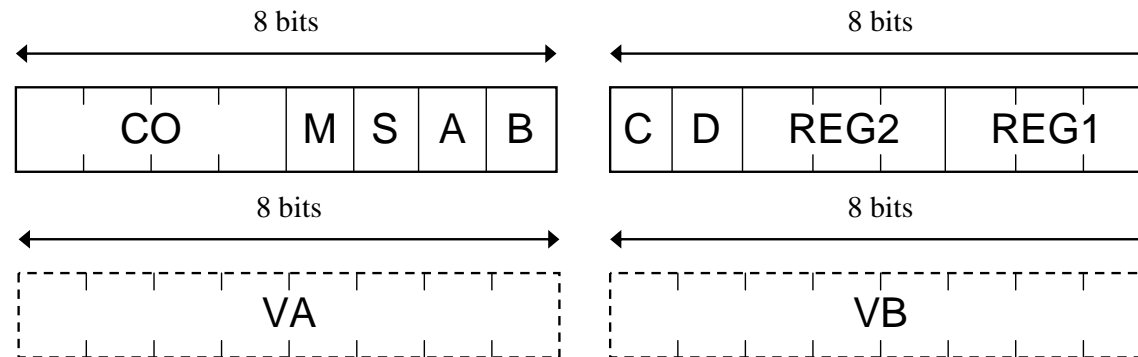
Modelo estructural del procesador VCore



Modelo funcional del procesador VCore



Formato de instrucción y modos de direccionamiento



CO: código de operación
 M: modo de operación
 S: tamaño de operandos
 ABCD: direccionamiento
 REG1,REG2: registros
 VA,VB: valor numérico

ABCD	Direccionamiento	ABCD	Direccionamiento
0000	Registro	1000	$DE \leftarrow (MP[(.REG2)])$
0001	Inmediato corto VA	1001	$DE \leftarrow (MP[(.REG2)]) ; REG2 += S$
0010	Inmediato largo VA:VB	1010	$DE \leftarrow (VA) + (.REG2)$
0011	$DE \leftarrow (VA)$	1011	$DE \leftarrow (VA:VB) + (.REG2)$
0100	$DE \leftarrow (VA:VB)$	1100	$DE \leftarrow (VA) + (.REG2) ; REG2 += S$
0101	$DE \leftarrow (MP[(VA:VB)])$	1101	$DE \leftarrow (VA:VB) + (.REG2) ; REG2 += S$
0110	$DE \leftarrow (.REG2)$	1110	$DE \leftarrow (MP[(VA) + (.REG2)]) ; REG2 += S$
0111	$DE \leftarrow (.REG2) ; REG2 += S$	1111	$DE \leftarrow (MP[(VA:VB) + (.REG2)]) ; REG2 += S$

Repertorio de instrucciones

Aritméticas (8 y 16 bits)		Control de flujo	
ADD,ADC	Suma normal y con acarreo	Saltos condicionales (cortos):	
SUB,SBC	Resta normal y con acarreo	BC,BNC : C BV,BNV : V	
CMP	Comparación	BN,BNN : N BZ,BNZ : Z	
NEG	Negación aritmética	BLT,BLE,BGT,BGE,BLS,BHI	
INC,DEC	Incremento y decremento	Saltos incondicionales:	
Lógicas (8 y 16 bits)		BR,JMP saltos cortos y largos	
AND	Operación lógica Y	CALL , RET llamadas a subrutinas	
OR	Operación lógica O	RETI retorno de interrupción	
XOR	Operación lógica O exclusivo	Operaciones con la pila (reg 7)	
NOT	Negación logica	PUSH,PUSHSR meter datos en pila	
Corrimiento de bits		POP,POPSR sacar datos de pila	
LSR,LSL	Corrimientos lógicos (ASL)	Otras operaciones	
ROR,ROL	Corrimientos circulares	IN , OUT E/S puertos de periféricos	
ASR	Corrimiento aritmético	CLC,CLV,CLI,STI,BRK,LD,ST...	

Indicadores de unidad aritmética y lógica (ALU)

Indicadores en instrucciones aritméticas

$$V \leftarrow A_{15} \& B_{15} \& \overline{R_{15}} \mid \overline{A_{15}} \& \overline{B_{15}} \& R_{15}$$

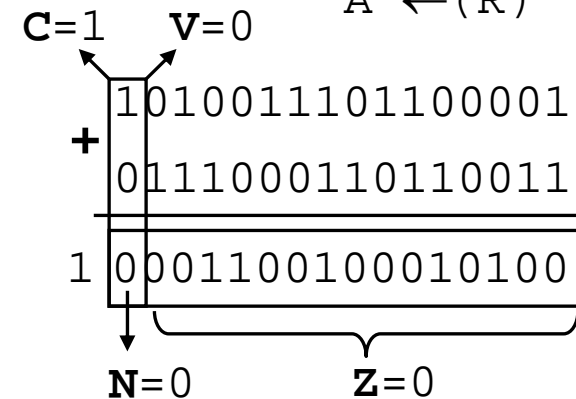
$$N \leftarrow R_{15}$$

$$Z \leftarrow \overline{R_{15}} \& \overline{R_{14}} \& \overline{R_{13}} \& \overline{R_{12}} \& \dots \& \overline{R_1} \& \overline{R_0}$$

$$C \leftarrow A_{15} \& B_{15} \mid B_{15} \& \overline{R_{15}} \mid \overline{R_{15}} \& A_{15}$$

ADD A, B : $R \leftarrow (A) + (B)$

$A \leftarrow (R)$



Indicadores en instrucciones lógicas

Ejemplo: operación AND A, B

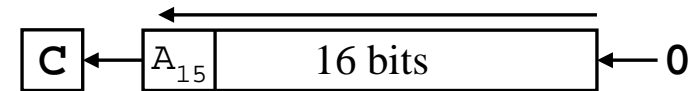
$$\begin{array}{r} \& 1010111100010110 \\ 0101001000011110 \\ \hline 0000001000010110 \end{array}$$

$$V \leftarrow 0$$

$$N \leftarrow R_{15} \quad Z \leftarrow \overline{R_{15}} \& \dots \& \overline{R_0}$$

Indicadores en corrimientos de bits

Ejemplo: operación ASL A



$$V \leftarrow R_{15} \oplus A_{15} \quad Z \leftarrow \overline{R_{15}} \& \dots \& \overline{R_0}$$

$$N \leftarrow R_{15} \quad C \leftarrow R_{15}$$

Notación de programas en ensamblador VCore

ORG: indica el origen de la compilación de código

Etiquetas: señalan posiciones de memoria con un nombre

Valores: pueden tener notación binaria, octal, decimal o hexadecimal

END: debe colocarse siempre al final de un fichero fuente.

Existen otras directivas del ensamblador:

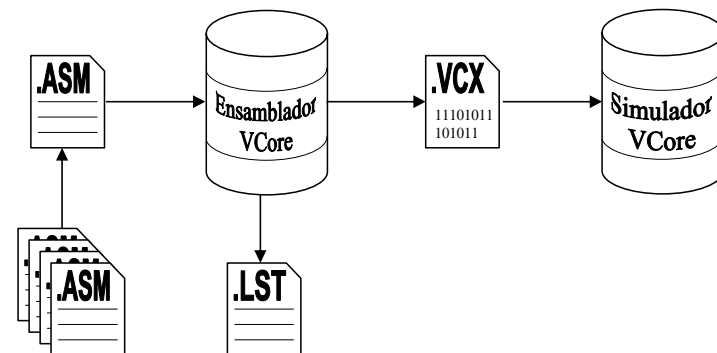
INCLUDE: para incluir otros archivos en el ensamblaje.

DB y DW: para reservar y declarar valores en zonas de memoria.

EQU: para crear cadenas que luego serán sustituidas por un valor textual.

ejemplo.asm

```
; Programa de ejemplo
                ORG    H'100
Inicio:         LD     .1, #100
                LD     .2, #D'200
BucleSinFin:    ADD    .1, .2 ; hacer algo
                BR     #BucleSinFin
                END     ; Final código
                NEG     .1    ; Ignorado !
```



Ejemplo programa VCore: suma en triple precisión

sumadp.asm

```
; Suma de doble precisión
                ORG    H'100
Inicio:        LD      .1,#PrimerSum
                LD      .2,#SegundoSum
                LD      .3,#Resultado
                LD      .6,/4[.1]
                LD      .5,/2[.1]
                LD      .4,[.1]
                ADD     .6,/4[.2]
                ADC     .5,/2[.2]
                ADC     .4,[.2]
                ST      .4,[.3++]
                ST      .5,[.3++]
                ST      .6,[.3++]
                HALT
PrimerSum:     DW      H'1B73,H'F435,H'89BC
SegundoSum:    DW      H'43C2,H'3121,H'9A4F
Resultado:     DW      ?,?,?
                END
```

ADD	+	H' 89BC	
		H' 9A4F	
		<hr/>	
		H' ①240B	
<hr/>			
ADC	+	H' F435	
	+	H' 3121	
	+	H' 0001	←
		<hr/>	
		H' ①2557	
<hr/>			
ADC	+	H' 1B73	
	+	H' 43C2	
	+	H' 0001	←
		<hr/>	
		H' 5F36	

H' 5F36 2557 240B