

# 6502 Assmbler in Forth (fig style)#

```
SCR # 81
0 ( FORTH-65 ASSEMBLER WFR-79JUN03 )
1 HEX
2 VOCABULARY ASSEMBLER IMMEDIATE ASSEMBLER DEFINITIONS
3
4 ( REGISTER ASSIGNMENT SPECIFIC TO IMPLEMENTATION )
5 E0 CONSTANT XSAVE DC CONSTANT W DE CONSTANT UP
6 D9 CONSTANT IP D1 CONSTANT N
7
8 ( NUCLEUS LOCATIONS ARE IMPLEMENTATION SPECIFIC )
9 ' (DO) 0E + CONSTANT POP
10 ' (DO) OC + CONSTANT POPTWO
11 ' LIT 13 + CONSTANT PUT
12 ' LIT 11 + CONSTANT PUSH
13 ' LIT 18 + CONSTANT NEXT
14 ' EXECUTE NFA 11 - CONSTANT SETUP
15
```

```
SCR # 82
0 ( ASSEMBLER, CONT. WFR-78OCT03 )
1 0 VARIABLE INDEX -2 ALLOT
2 0909 , 1505 , 0115 , 8011 , 8009 , 1DOD , 8019 , 8080 ,
3 0080 , 1404 , 8014 , 8080 , 8080 , 1COC , 801C , 2C80 ,
4
5 2 VARIABLE MODE
6 : .A 0 MODE ! ; : # 1 MODE ! ; : MEM 2 MODE ! ;
7 : ,X 3 MODE ! ; : ,Y 4 MODE ! ; : X) 5 MODE ! ;
8 : )Y 6 MODE ! ; : ) F MODE ! ;
9
10 : BOT ,X 0 ; ( ADDRESS THE BOTTOM OF THE STACK *)
11 : SEC ,X 2 ; ( ADDRESS SECOND ITEM ON STACK *)
12 : RP) ,X 101 ; ( ADDRESS BOTTOM OF RETURN STACK *)
13
14
15
```

```
SCR # 83
0 ( UPMODE, CPU WFR-78OCT23 )
1
2 : UPMODE IF MODE @ 8 AND 0= IF 8 MODE +! THEN THEN
3 1 MODE @ OF AND -DUP IF 0 DO DUP + LOOP THEN
4 OVER 1+ @ AND 0= ;
5
6 : CPU <BUILDS C, DOES> C@ C, MEM ;
7 00 CPU BRK, 18 CPU CLC, D8 CPU CLD, 58 CPU CLI,
8 B8 CPU CLV, CA CPU DEX, 88 CPU DEY, E8 CPU INX,
9 C8 CPU INY, EA CPU NOP, 48 CPU PHA, 08 CPU PHP,
10 68 CPU PLA, 28 CPU PLP, 40 CPU RTI, 60 CPU RTS,
11 38 CPU SEC, F8 CPU SED, 78 CPU SEI, AA CPU TAX,
12 A8 CPU TAY, BA CPU TSX, 8A CPU TXA, 9A CPU TXS,
13 98 CPU TYA,
14
15
```

SCR # 84

```

0 ( M/CPU, MULTI-MODE OP-CODES WFR-79MAR26 )
1 : M/CPU <BUILDS C, , DOES>
2 DUP 1+ @ 80 AND IF 10 MODE +! THEN OVER
3 FFOO AND UPMODE UPMODE IF MEM CR LATEST ID.
4 3 ERROR THEN C@ MODE C@
5 INDEX + C@ + C, MODE C@ 7 AND IF MODE C@
6 OF AND 7 < IF C, ELSE , THEN THEN MEM ;
7
8 1C6E 60 M/CPU ADC, 1C6E 20 M/CPU AND, 1C6E C0 M/CPU CMP,
9 1C6E 40 M/CPU EOR, 1C6E A0 M/CPU LDA, 1C6E 00 M/CPU ORA,
10 1C6E E0 M/CPU SBC, 1C6C 80 M/CPU STA, 0D0D 01 M/CPU ASL,
11 0C0C C1 M/CPU DEC, 0C0C E1 M/CPU INC, 0D0D 41 M/CPU LSR,
12 0D0D 21 M/CPU ROL, 0D0D 61 M/CPU ROR, 0414 81 M/CPU STX,
13 0486 E0 M/CPU CPX, 0486 C0 M/CPU CPY, 1496 A2 M/CPU LDX,
14 0C8E A0 M/CPU LDY, 048C 80 M/CPU STY, 0480 14 M/CPU JSR,
15 8480 40 M/CPU JMP, 0484 20 M/CPU BIT,

```

SCR # 85

```

0 ( ASSEMBLER CONDITIONALS WFR-79MAR26 )
1 : BEGIN, HERE 1 ; IMMEDIATE
2 : UNTIL, ?EXEC >R 1 ?PAIRS R> C, HERE 1+ - C, ; IMMEDIATE
3 : IF, C, HERE 0 C, 2 ; IMMEDIATE
4 : THEN, ?EXEC 2 ?PAIRS HERE OVER C@
5 IF SWAP ! ELSE OVER 1+ - SWAP C! THEN ; IMMEDIATE
6 : ELSE, 2 ?PAIRS HERE 1+ 1 JMP
7 SWAP HERE OVER 1+ - SWAP C! 2 ; IMMEDIATE
8 : NOT 20 + ; ( REVERSE ASSEMBLY TEST )
9 90 CONSTANT CS ( ASSEMBLE TEST FOR CARRY SET )
10 DO CONSTANT O= ( ASSEMBLER TEST FOR EQUAL ZERO )
11 10 CONSTANT O< ( ASSEMBLE TEST FOR LESS THAN ZERO )
12 90 CONSTANT >= ( ASSEMBLE TEST FOR GREATER OR EQUAL ZERO )
13 ( >= IS ONLY CORRECT AFTER SUB, OR CMP, )
14
15

```

SCR # 86

```

0 ( USE OF ASSEMBLER WFR-79APR28 )
1 : END-CODE ( END OF CODE DEFINITION *)
2 CURRENT @ CONTEXT ! ?EXEC ?CSP SMUDGE ; IMMEDIATE
3
4 FORTH DEFINITIONS DECIMAL
5 : CODE ( CREATE WORD AT ASSEMBLY CODE LEVEL *)
6 ? EXEC CREATE (COMPILE) ASSEMBLER
7 ASSEMBLER MEM !CSP ; IMMEDIATE
8
9 ( LOCK ASSEMBLER INTO SYSTEM )
10 ' ASSEMBLER CFA ' ;CODE 8 + ! ( OVER-WRITE SMUDGE )
11 LATEST 12 +ORIGIN ! ( TOP NFA )
12 HERE 28 +ORIGIN ! ( FENCE )
13 HERE 30 +ORIGIN ! ( DP )
14 ' ASSEMBLER 6 + 32 +ORIGIN ! ( VOC-LINK )
15 HERE FENCE !

```