

Lecture 5:

Unconditional Branches

Today's Goals

- Learn hexadecimal addition and subtraction.
- Use unconditional branches and jump instructions.

Hexadecimal Addition and Subtraction

- For the most part, adding and subtracting in hexadecimal is performed like decimal.
- Only difference is carries or borrows 16 instead of 10.
- Examples

$$\text{\$}2367 + \text{\$}5FD6 = \text{\$}833D \qquad \text{\$}AC22 - \text{\$}1EE8 = \text{\$}8D3A$$



Extending Hexadecimal Numbers

- When a shorter number is added to or subtracted from a longer number, the shorter number must be extended to the same number of digits as the longer number

Always extend by adding 0's

Repeating the sign bits.
= Adding 'F's or '0's

- Unsigned:

$\$2357 + \$D6$

Signed:

$\$2357 + \$D6$

$\$2357 + \$6D$



Truncating Hexadecimal Numbers

- In microcomputers, it is common practice to represent values using a fewer number of bits to save both time and space.
- Shortly, we will see the need to represent the value of a two-byte hexadecimal numbers as a one-byte value if possible.

Remove only leading 0s. The remaining number is always valid.

Remove leading 0s and have a positive value or
Remove leading Fs and have a negative value.

- Unsigned

\$00F5

\$10EC ✖

Signed:

\$0045

\$00F5 ✖

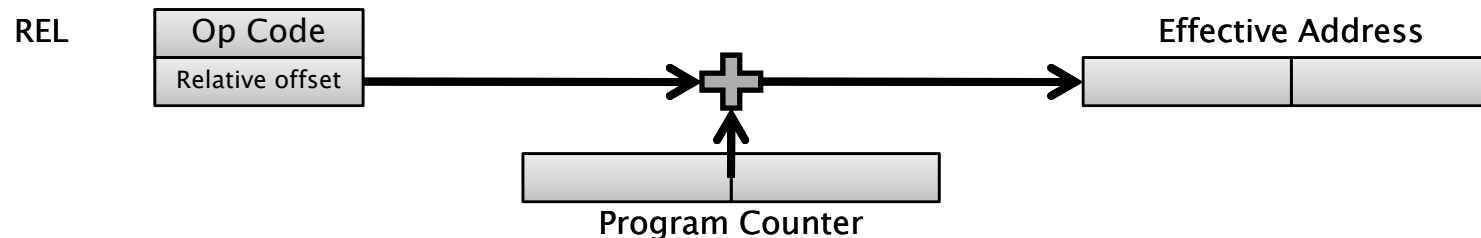
\$FFD1

\$FF66 ✖

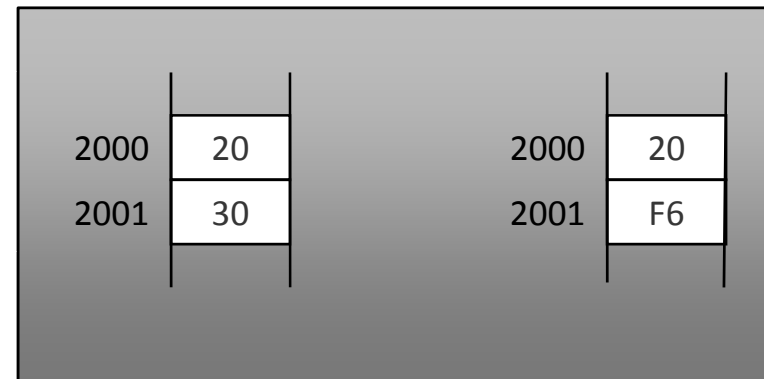
\$F280 ✖

Relative Addressing (REL)

- Effective Address:
 - Add the operand as a signed number to the value in the PC.



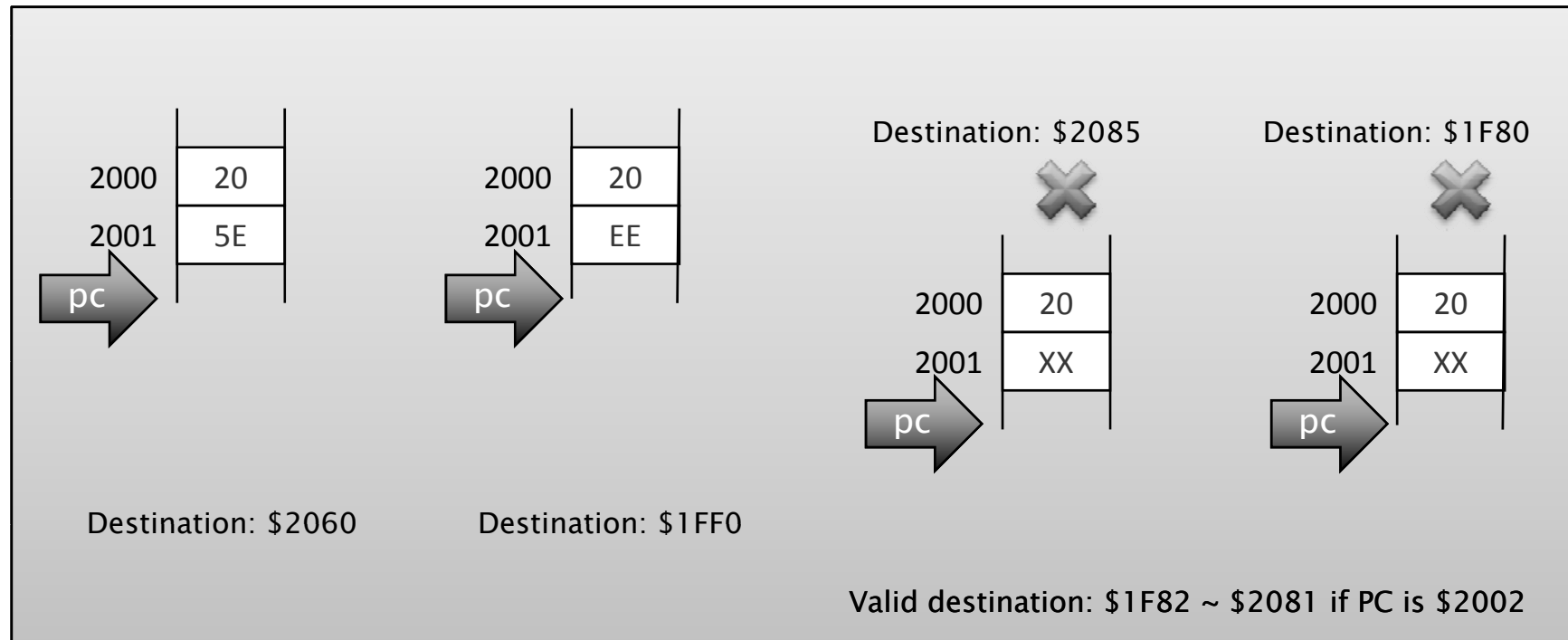
- The effective address is loaded into the PC, and the program executes from the new address
- Examples
 - BRA \$30
 - Branch always to the instruction 30h bytes forward.
 - BRA -10
 - Branch always to the instruction 10 bytes backwards



Calculating Branch Destinations

Valid range: $-\$80 \sim \$7F$

- 'Branch' means changing a value of the program counter in the point of view of the microprocessor.
- The destination address can be calculated by adding the operand (either + or -0) to the value of the current PC.



Branch and Jump

Instructions

- BRA
 - Branch always
 - Only uses relative addressing (REL) with one-byte operands
- LBRA
 - Long Branch always
 - Only uses relative addressing (REL) with two-byte operands
- JMP
 - Jump
 - Uses extended(EXT) or index addressing (IDX)

Questions?

Wrap-up

What we've learned

- Hexadecimal addition and subtraction
- Unconditional branch instructions

What to Come

- Instructions for conditional branches
- HCS12 Assembly language