

## Experiment 4-A

# ADDRESSING MODES

### OBJECTIVE

Learn about different types of addressing modes on eZ8 CPU and experiment on each method.

### 1. Addressing Modes

The eZ8 CPU provides six addressing modes:

- Register (R)
- Indirect Register (IR)
- Indexed (X)
- Direct (DA)
- Relative (RA)
- Immediate Data (IM)

NOTATION	DESCRIPTION	OPERAND	RANGE
R	Register	Reg	Reg. represents a number in the range of 00H to FFH
IR	Indirect Register	@Reg	Reg. represents a number in the range of 00H to FFH
X	Indexed	#Index	The register or register pair to be indexed is offset by the signed Index value (#Index) in a +127 to -128 range.
DA	Direct Address	Addr	Addr. represents a number in the range of 0000H to FFFFH
RA	Relative Address	X	X represents an index in the range of +127 to -128 which is an offset relative to the address of the next instruction
IM	Immediate Data	#Data	Data is a number between 00H to FFH

Create a new project in Z8 Encore! IDE and add a new empty asm file to the project. Then write and study the following code:

```
vector reset = startup
org %1000

startup:
srp #00

; A little housekeeping
ld r0, %%0f
loop:
    ld @r0, %%0
    dec r0
    jr nz, loop





; Register Addressing
ld 00h, %%01
ld 05h, %%ef

; Indirect Register Addressing
ld @00h, %%08
ld @01h, %%10

; Indexed Addressing
ld 03h+3, %%dd
ld 07h-4, %%ee

; Direct Addressing
ld 0000h, %%02
ld 0004h, %%0f

; Relative Addressing
ld 0001h+4, %%04
ld 000fh-13, %%01
```

Build the project by clicking the **Rebuild All** icon (  ) on toolbar. Wait for the building process to be completed. Then click the Reset icon (  ) to connect and download the code to the development board. When the debugging toolbar appears, click Memory icon (  ) to view the device memory. In the memory window, set Memory Space to **Rdata** and Address to **R#000**. Keep clicking on the Step Into icon (  ) and watch and understand how the memory window changes with each assembly instruction.

## EXPERIMENTAL WORK

Using all the methods explained above, create your own complete assembly code which will display your own name and surname in the memory window.

As an example;

```
ld 00h,    %%45 ; E
ld 00h+1,  %%46 ; F (using indexed addressing)
ld 02h,    00h ; E
```

**TIP:** Hexadecimal values of English alphabet letters start at 41 (A) and end at 5A (Z). Space character's hexadecimal value is 20. You may find the complete set below:

<b>A</b>	<b>41</b>	<b>E</b>	<b>45</b>	<b>I</b>	<b>49</b>	<b>M</b>	<b>4D</b>	<b>Q</b>	<b>51</b>	<b>U</b>	<b>55</b>	<b>Y</b>	<b>59</b>
<b>B</b>	<b>42</b>	<b>F</b>	<b>46</b>	<b>J</b>	<b>4A</b>	<b>N</b>	<b>4E</b>	<b>R</b>	<b>52</b>	<b>V</b>	<b>56</b>	<b>Z</b>	<b>5A</b>
<b>C</b>	<b>43</b>	<b>G</b>	<b>47</b>	<b>K</b>	<b>4B</b>	<b>O</b>	<b>4F</b>	<b>S</b>	<b>53</b>	<b>W</b>	<b>57</b>	<b>space</b>	<b>20</b>
<b>D</b>	<b>44</b>	<b>H</b>	<b>48</b>	<b>L</b>	<b>4C</b>	<b>P</b>	<b>50</b>	<b>T</b>	<b>54</b>	<b>X</b>	<b>58</b>		