Static and Dynamic Relocation

Introduction
This shows the basic hardware instruction cycle for a machine that uses static relocation and for one that uses dynamic relocation.

Static Relocation
Static relocation refers to address transformations being done before execution of a program begins. A typical hardware instruction cycle looks like this:

```plaintext
while true do begin
  w := M[inst_cotr]; (* fetch instruction *)
  oc := Opcode(w);
  adr := Address(w);
  inst_cotr := inst_cotr + 1;
  case oc of
    1: reg := reg + M[adr]; (* add *)
    2: M[adr] := reg;     (* store *)
    3: inst_cotr := adr;  (* branch *)
    ...
  end
end (* loop *)
```

Dynamic Relocation
Dynamic relocation refers to address transformations being done during execution of a program. In what follows, the function NLmap (for Name Location map) maps the relocatable (virtual) address va given in the program into the real (physical) storage address pa:

```plaintext
pa := NLmap(va)
```

So, a typical hardware instruction cycle looks like this:

```plaintext
while true do begin
  w := M[NLmap(inst_cotr)]; (* fetch instruction *)
  oc := Opcode(w);
  adr := Address(w);
  inst_cotr := inst_cotr + 1;
  case oc of
    1: reg := reg + M[NL_map adr]; (* add *)
    2: M[NL_map adr] := reg;     (* store *)
    3: inst_cotr := NL_map adr;  (* branch *)
    ...
  end
end (* loop *)
```