Prog1ram 4

ARM

Print and use
the specification document
on the class web site

See syllabus for assignment type
individual or team

Open input *key.in* & output *key.out*

Read a line of ASCII text (00-7Fh)

For each character in the input string :
* **uppercase letter move to output string**
* **lowercase letter convert to upper case move to output string**
* **blank move it to output string**
* **anything else throw away ... including any control characters**

Write output string + CRLF output file

Repeat till *read string SWI* returns EOF

Close input and output file and halt

Step 0. Install the ARMSim Assembler / Simulator

Step 1. Create a design

Use sample programs
(hello.s copystr.s copyfile.s)
as models

Step 2. Code your solution

Retrieve the grading system packed in a self-extracting file named *unpack.exe*.

Save in DOSBox \P23X\ARM directory

In DOSBox type: *unpack*

- **use sample code as models**
- **name your file armkey.s**

Step 3. Test and debug your solution.

- **Use the sample file key.in (modify it for additional tests)**
- **Use ARMSim to read, assemble and run your program**
- **Your program's output is *key.out***
- **Verify *key.out* is correct**

Step 4. Grade

Grade your program following the instructions in the specification

Step 5. Submit your assignment

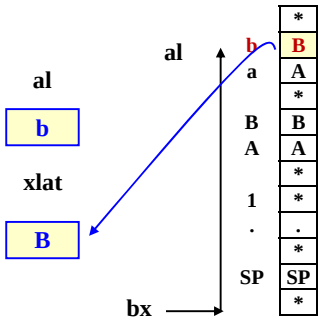
Electronically submit the file

arm.ans

created by the grading system.

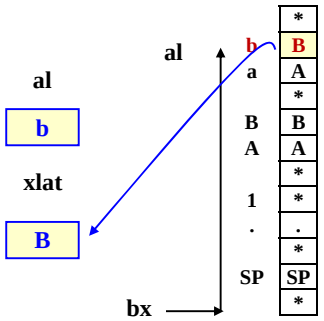
Grade based on
- **Correct answers**
- **Number of instructions written**
- **Documentation**

Design Idea



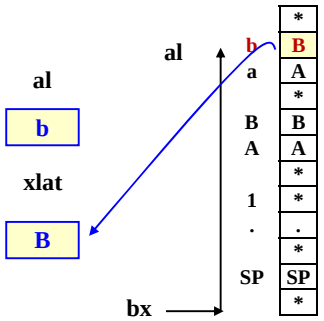
- **8086 KEY used XLAT**

Design Idea



- **8086 KEY used XLAT**
- **XLAT is a form of indirect addressing using two registers ... bx and al**

Design Idea



- **8086 KEY used XLAT**
- **XLAT is a form of indirect addressing using two registers ... bx and al**
- **ARM supports indirect addressing using two registers**