ELEC6234
Embedded Processor Synthesis

PicoMIPS architecture
picoMIPS – our case study

- Inspired by MIPS, but not a version of MIPS
- Architecture suited for efficient synthesis on FPGAs with synchronous RAM blocks
- User specifies the width of the data bus n, typically 8, 16, 32, 64 bits.
- User specifies the instruction set to be implemented in the decoder

pMIPS version 1 – no RAM, no branches
picoMIPS instruction groups

- ALU instructions operate on registers only, not RAM
  - ADD, SUB, AND, OR, XOR, NOR
    - 2 register format: \( \text{ADD} \ %d, \ %s \rightarrow \ %d = %d + %s \)
  - ADDI, SUBI, ANDI, ORI, Xori, NORI
    - registers + immediate \( \text{ADDI} \ %d, \ %s \text{imm} \rightarrow \ %d = %s + \text{imm} \)
  - MUL, shift instructions SLL, SRL, SRA (will not be included in our case study)

- Program flow control
  - J address - unconditional absolute jump
  - BEQ \%d, %s address – conditional relative branch: branch if \%d-%s == 0
  - Other conditional branches:
    - BNE branch if // Z=0,
    - `define BGE // ~ N ; BLT %d,%s == BGE %s,%d
    - `define BLO // C=1 (lower, unsigned), BHI %d,%s

- RAM related instructions
  - LW \%d, [address]—load word from RAM into register \%d
  - SW %s, [address] – store word from register %s into RAM

- Others
  - Eg. NOP; no operation
  - Subroutine calls and returns, stack instructions – not implemented in our case study

- General approach when synthesising on an FPGA – the decoder should only implement the instructions used by the program stored in the program memory
  - Hardware efficient approach
Translating C statements to picoMIPS machine code
- if statement

- if (a==0) b=1; else b=2;

```
def a=20    ; a is register 20
.def b=21    ; b is register 21
BNE %a, %0, Lelse ; if a!=0 goto Lelse, nb. reg %0 is always 0
                ; load immediate value to b:
LDI %b, 1     ; load 1 to %b (ADDI %b, %0, 1)
J Lend        ; jump to Lend
Lelse:        ...
    LDI %b,2  ; actual code: ADDI %b, %0, 2
Lend:
```
Translating C statements to picoMIPS machine code
- while statement

- \( i=0; n=10; a=10; \text{ while } (i<=n) \{ a += a; i++; \} \)

```
.def i=16
.def n=17
.def a=18
LDI %i,0
LDI %n,10
LDI %a,10

Loop:
    BLO %n, %i, Lend ; branch to exit loop if Carry set, i.e. n<i
    ADD %a, %a ; a = a+a
    ADDI %i, %1 ; i = i + 1
    J Loop ; jump to Loop

Lend:        …
```
Quick exercise:

- Translate the following ‘for’ statement to picoMIPS assembler

```mips
.def  i=16
.def  n=17

for(i=0;i<=n;i++);
```

...
Steps in embedded processor synthesis (picoMIPS case study)

• Develop and test ALU with typical operations: ADD, SUB, etc.
• Develop General Purpose Registers (GPR) and test
• Develop a simple Instruction Decoder and test with ALU and GPR
  – in the decoder implement NOP instruction, ALU instructions, LDI
• Develop Program Memory
• Develop Program Counter
• Enhance Instruction Decoder with J and typical conditional branches, e.g. BEQ, BLO etc.
• Basic processor core is functional at this point
• Possible enhancements:
  – Add data RAM
  – RAM related instructions, e.g. LD, ST
  – Develop an I/O port, and implement in decoder IN and OUT instructions
  – Add ALU hardware to support MUL instruction and shift instructions
  – Subroutine calls and returns
• Your coursework assignment:
  – Modify the picoMIPS to implement a prescribed program
  – Write a formal report
  – Challenge: the smallest possible implementation on Altera Cyclone IV