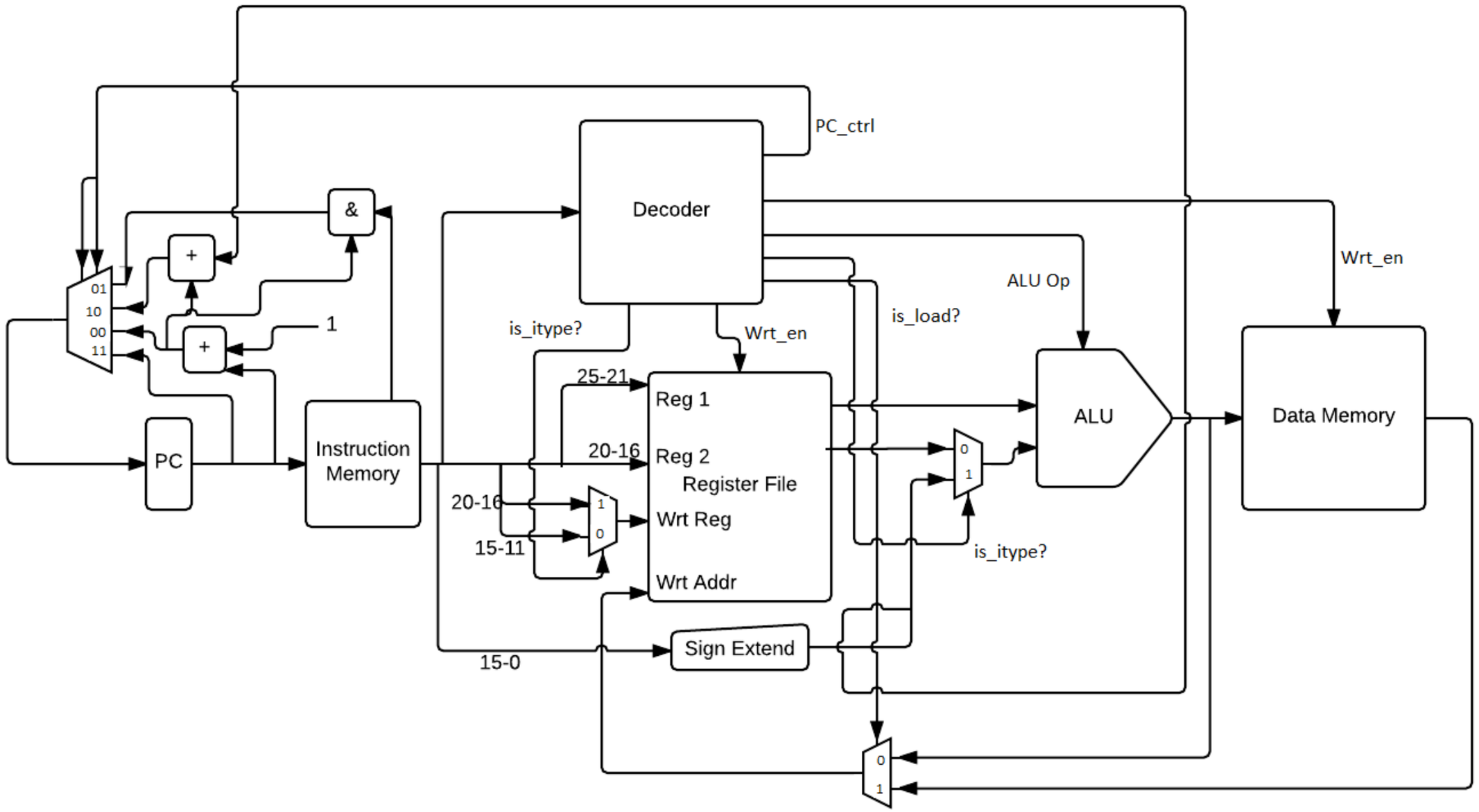
Processor

by

Arora Siddhartha
Bou Khalil Carl
Kwan Kevin

Moyseyev Dmytro
Plaudis Roberts
Sharma Anusha

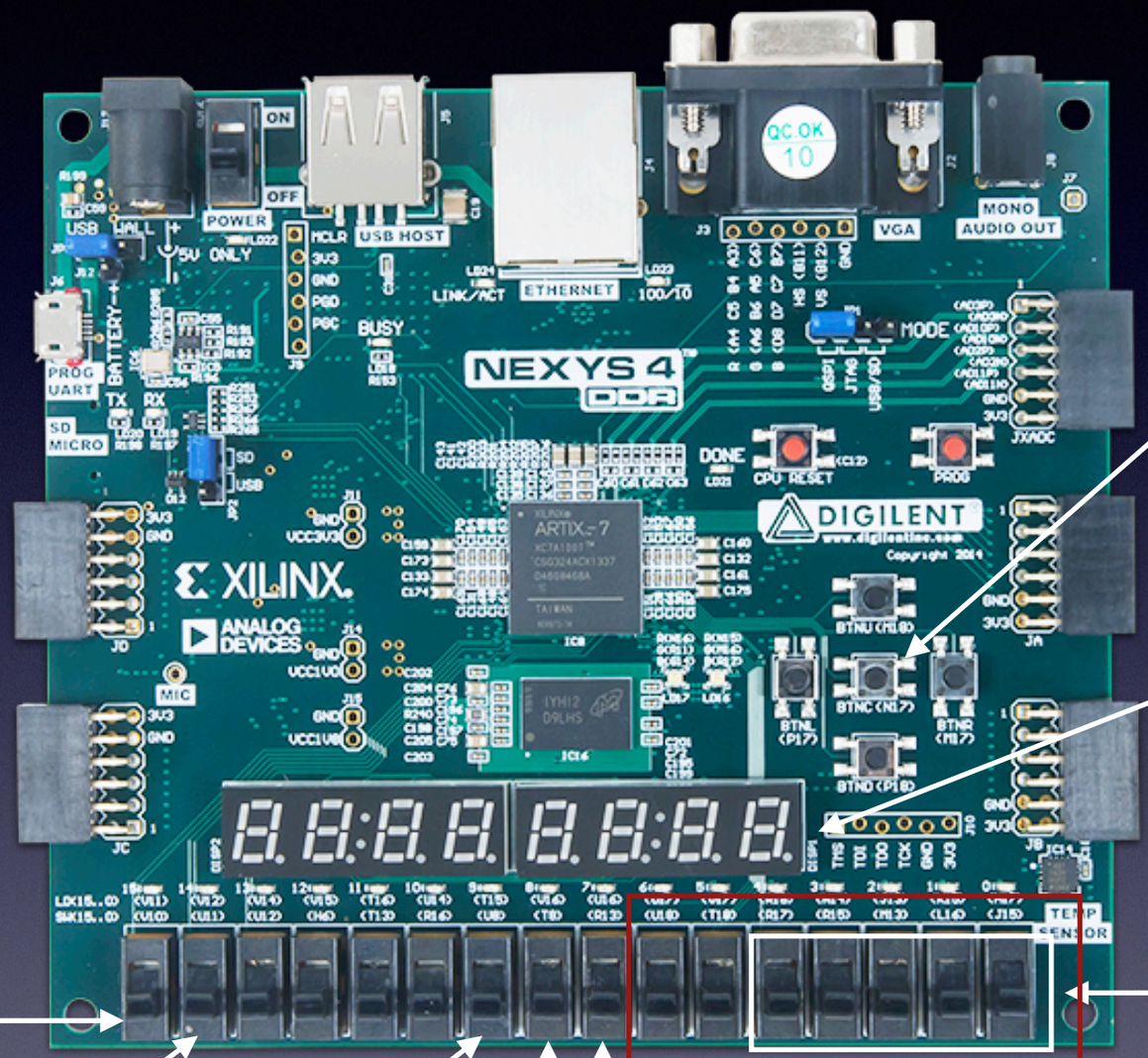
Block Diagram



Steps

- Implemented each block of the block diagram as a separate component
- Tested each individual component
- Added them together in a top level module
- Added multiplexers and logic for next PC value calculation
- Added single stepping
- Added inputs and outputs (Interface)

Interface



Clear

Display a 32-bit value depending on the value of switches 7 and 8

Simulates clock when single stepping enabled

Enables (1) single stepping or disables (0) it

Encryption (1) or decryption (0)

“00” shows the instruction
“01” shows content of RF
“10” shows content of DM
“11” shows PC value

What register to show the content of

What DM location to show the content of

RC5

RC5 or “Rivest Cipher” is a symmetric key block cipher notable for its simplicity. It was designed by Ronald Rivest in 1994. The RC5 cipher designed by us has a block size of 64 bits, key size of 128 bits and operates for 78 rounds. The key feature of RC5 is the use of data dependent rotation.

Key expansion

```
S[0] = 0xB7E15163 (Pw)
for i=1 to 25 do
   S[i] = S[i-1]+ 0x9E3779B9 (Qw)

for i = b - 1 downto 0 do
   L[i/u] = (L[i/u] <<< 8) + K[i];

do 3*max(t, c); t=26, c=4
   A = S[i] = (S[i] + A + B) <<< 3;
   B = L[j] = (L[j] + A + B) <<< (A + B);
   i = (i + 1) mod (t);
   j = (j + 1) mod (c);
```

Encryption

```
A = A + S[0];
B = B + S[1];

for i = 1 to 12 do
   A = ((A xor B) <<< B) + S[2×i];
   B = ((B xor A) <<< A) + S[2×i + 1];
```

Decryption

```
for i = 12 down to 1 do
   B = ((B - S[2×i +1]) >>> A) xor A;
   A = ((A - S[2×i]) >>> B) xor B;

B = B - S[1];
A = A - S[0];
```

RC5 on processor

- Translated algorithms to assembly code
- Created C code to translate from assembly to binary and hexadecimal
- Translated assembly code to machine code
- Tested the individual parts of rc5
- Assembled all the parts in one assembly code
- Translated to machine code
- Tested the complete RC5

RC5 on processor

- User input (user key, A and B) stored in the data memory beforehand
- L array and S array stored in the data memory
- Encryption or decryption results (A’ and B’) stored in the data memory
- Value to choose between encryption and decryption stored in data memory but can be changed by the user using switch 9

Data Memory

0	1	2	3	4	5	6	7	8	9
ukey(3)	ukey(2)	ukey(1)	ukey(0)	L[0]	L[1]	L[2]	L[3]	Pw	Qw

10	11	12	13	14	15	16	17	18	19
skey[0]	skey[1]	skey[2]	skey[3]	skey[4]	skey[5]	skey[6]	skey[7]	skey[8]	skey[9]

20	21	22	23	24	25	26	27	28	29
skey[10]	skey[11]	skey[12]	skey[13]	skey[14]	skey[15]	skey[16]	skey[17]	skey[18]	skey[19]

30	31	32	33	34	35	36	37	38	39
skey[20]	skey[21]	skey[22]	skey[23]	skey[24]	skey[25]	A	B	A’	B’

40
Enc