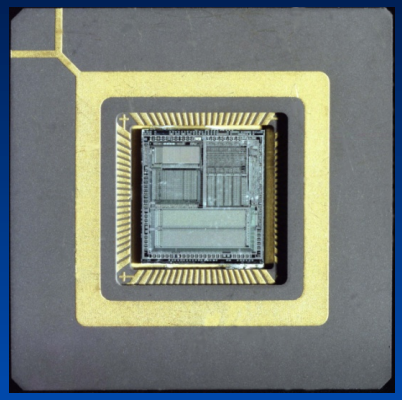


T42 – Transputer Design in FPGA

Year-One Design Status Report

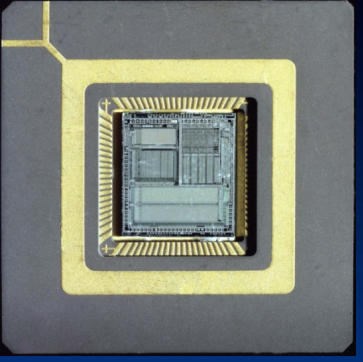
Uwe MIELKE [a] and Martin ZABEL [b],
in collaboration w/ Michael BRUESTLE [c]

[a] *Electronics Engineer, Dresden, Germany,* uwe.mielke@t-online.de

[b] *Institut of Computer Engineering, Technische Universität Dresden, Germany,* martin.zabel@tu-dresden.de

[c] *Electronics Engineer, Vienna, Austria,* michael_bruestle@yahoo.com

Communicating Process Architectures 2015



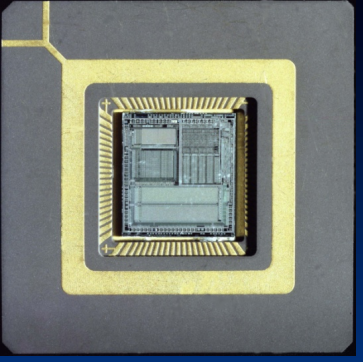
T42 in FPGA @ CPA 2015

Abstract:

This fringe session will present the current status of our still ongoing IMS-T425 compatible Transputer design in FPGA.

Data path and control path are in a stable working condition. Fetch unit and a basic system control unit are almost functional. Small instruction sequences can be executed from 8Kbyte memory already.

Some details arround scheduler micro-code will be discussed.



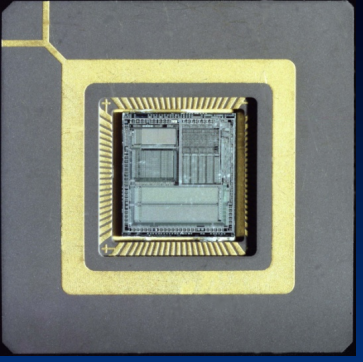
T42 in FPGA @ CPA 2014

Our Motivation:

- Overcome absence of CSP (Transputers and Occam) in public
- Provide a free, IMS-T425 binary compatible, open source VHDL
- Many T42 cores fit into small FPGA e.g. 2 in XC6S-LX9 →16+ in XC2S-LX100
- VHDL is easy to download, easy to improve … let ‘s enhance it !
- **Computer Engineering Students need toys to play with !**
- TU Dresden has experience with own Java MultiCore in FPGA

My (U.M.) personal motivation:

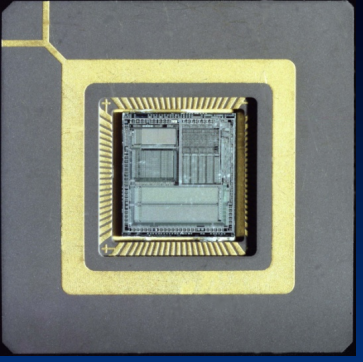
- I bunched into concurrency in 1983 … my diploma thesis: „a RTOS for Z80“
- I ‘m addicted to transputers since 1984 = concurrency elegance in hardware !



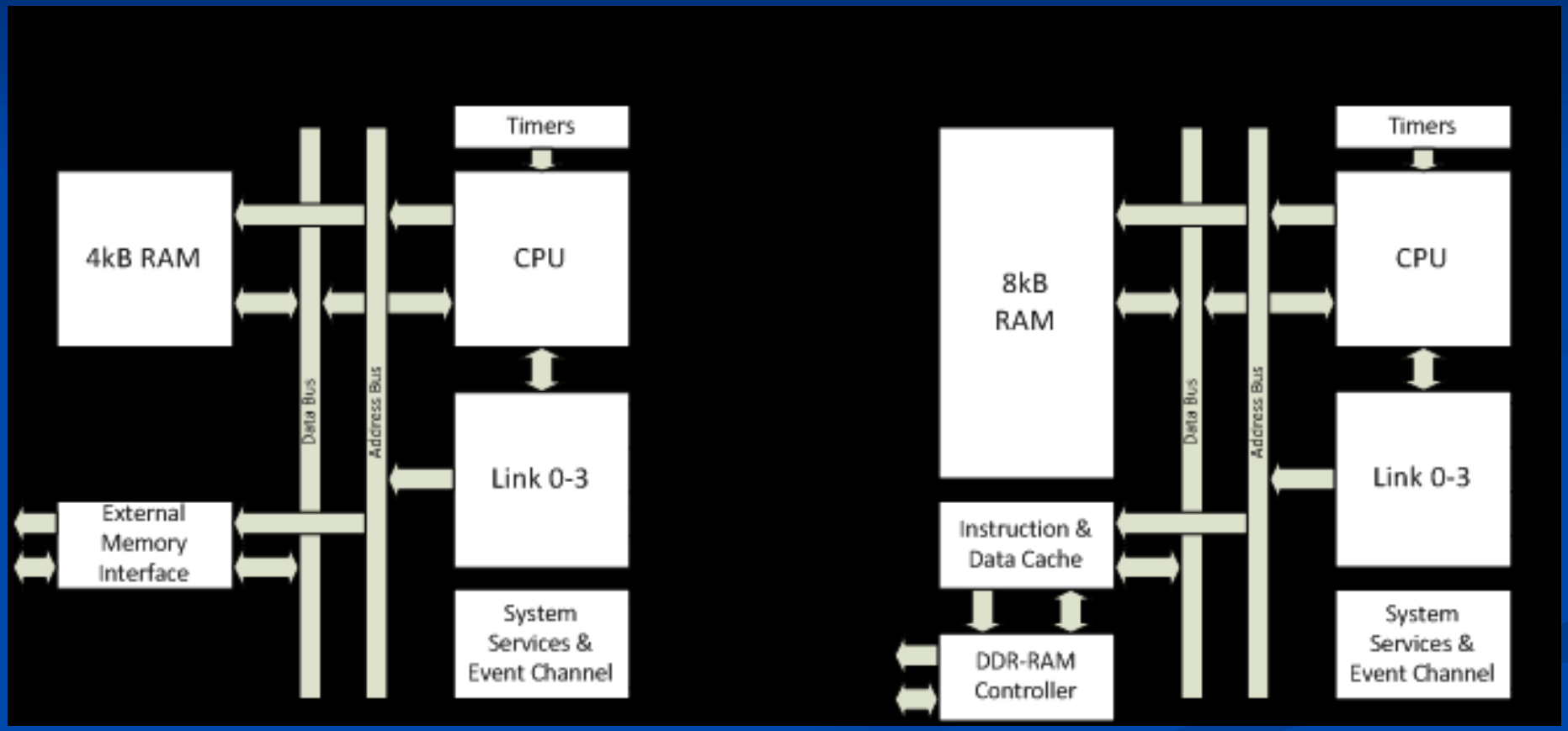
T42 Achievements

- T42 Project started May ‘2013 – VHDL Design started Jan ‘2014
- Data path and control path (1st concept) working … Apr ‘2014
- Microcode Assembler (12 AWK scripts) completed … Jan ‘2015
- ~50 simple OpCodes implemented, datapath extended Apr ‘2015
- Pipeline running (from 8 byte prefetch buffer) … May ‘2015
- onChip memory added (ldnl, stnl, …) and verified … Jun ‘2015
- Prefetch state machine + Iptr-Incrementor verified … Jul ‘2015
- System control unit, status bits, more flags added *… Aug ‘2015

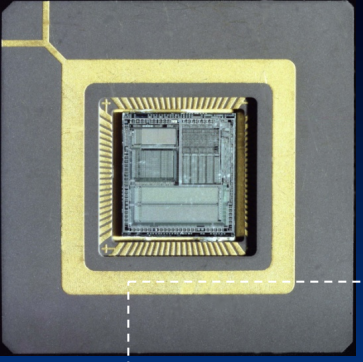
i.e. core infrastructure is almost complete, but … still * t.b. verified



IMS-T425 vs T42



Note: Sys Services take care of external pins & clock – Sys Control is part of CPU.



Remark: Blocks in red still N/A.

T42 VHDL Top View

T42cpu_all_top (structural)

SysPath:
-SysControl, Sbits, Timer, SysService

T42_cpu_constpkg

Ctrl2Data (structural) ← Pipeline

CtrlPath:
-uCodeROM
-Idecode
-Oreg
-Iptr (+Inc)
-PreFetch

DataPath:
-ABCDEreg
-ALU X+Y=Z
-Wptr
-Pointers
-ConstBox
-DataOutBus

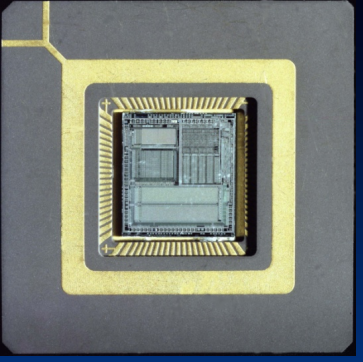
LinkPath:

MemPath:
-MemIF
-MemMain
(2kx32)

-DCache
-ICache
-eMemIF

MemLPDDR
(on FPGA board)

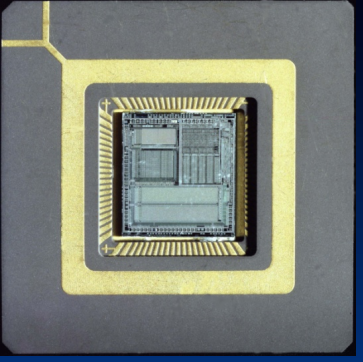
MemDDR2
(on FPGA board)



SW development takes longer than HW … about half a year effort :

uCode Asm

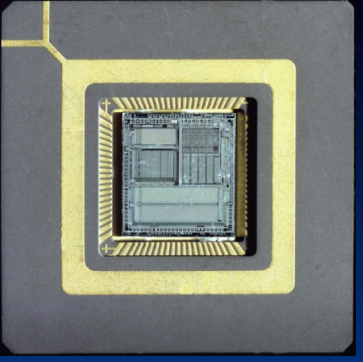
```
1.1 PRE PROCESSING - READ BIT POSITIONS OF MICRO-OP's
1.2 PRE PROCESSING - READ MICRO-OP IDs AND CODING
2.1 PRIMARY PROCESSING - ASSEMBLE MICROWORDs
2.2 SECONDARY PROCESSING - SORT MICROWORD COLUMNs
2.3 SECONDARY PROCESSING - SORT MICROWORD ROWs
2.4 SECONDARY PROCESSING - ALLOCATE FIXED ADDRESSES
2.5 SECONDARY PROCESSING - TABULATE BRANCH CAPABILITIES
2.6 SECONDARY PROCESSING - ALLOCATE JUMP+BRANCH LABELS
2.5 SECONDARY PROCESSING - CALCULATE ROMFEEDBAK ADDR
3.1 POST PROCESSING - BUILD ROM (BINARY FORMAT)
3.2 POST PROCESSING - WRITE HEX ROM
3.3 POST PROCESSING - BUILD uCodeROM
    CALL XILINX DATA2MEM
```



MicroWord

- T42: currently still … 96bit (about ~30 signals)
- T425 seems to have >100bit (uCodeROM ~60kBit)

```
ENTRYVALID;   95;   95;   -- 1 bit
NEXTACTION;   94;   94;   -- 1 bit
  __GAP_00__;   93;   93;   ++ 1
S_BIT_MODE;   91;   92;   -- 2 bit
S_BIT_FROM;   87;   90;   -- 4 bit
  __GAP_01__;   86;   86;   ++ 1
I_PTR_FROM;   84;   85;   -- 2 bit
W_PTR_FROM;   83;   83;   -- 1 bit
X_BUS_FROM;   80;   82;   -- 3 bit
Y_BUS_FROM;   77;   79;   -- 3 bit
MADDR_FROM;   75;   76;   -- 2 bit
A_REG_FROM;   72;   74;   -- 3 bit
A_SHIFT_IN;   70;   71;   -- 2 bit
B_REG_FROM;   67;   69;   -- 3 bit
B_SHIFT_IN;   65;   66;   -- 2 bit
CARRY_FROM;   63;   64;   -- 2 bit
CARRY_MODE;   62;   62;   -- 1 bit
C_REG_FROM;   59;   61;   -- 3 bit
C_SHIFT_IN;   57;   58;   -- 2 bit
CMUX1_FROM;   54;   56;   -- 3 bit
CMUX0_FROM;   51;   53;   -- 3 bit
  __GAP_02__;   50;   50;   ++ 1
D_REG_FROM;   47;   49;   -- 3 bit
O_REG_FROM;   45;   46;   -- 2 bit
MADDR_MODE;   43;   44;   -- 2 bit
MDATA_MODE;   41;   42;   -- 2 bit
Z_FROM_ALU;   35;   40;   -- 6 bit
E_REG_FROM;   32;   34;   -- 3 bit
E_SHIFT_IN;   30;   31;   -- 2 bit
MDATA_FROM;   27;   29;   -- 3 bit
POINT_FROM;   23;   26;   -- 4 bit
POINT_MODE;   22;   22;   -- 1 bit
  __GAP_03__;   17;   21;   ++ 5
CONST_FROM;   12;   16;   -- 5 bit
  __GAP_04__;    9;   11;   ++ 3
ROMFEEDBAK;    0;    8;   -- 9 bit
```

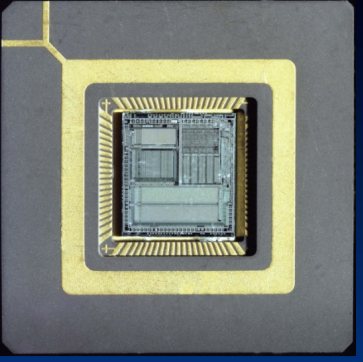


INMOS Patent Research

Scheduler, Timer, Link investigations based on:

- US-Pat-4989133 – INMOS 29Jan1991
  System for executing time dependent processes
- US-Pat-4783734 – INMOS 08Nov1988
  Computer with variable length process communication
- US-Pat-4794526 – INMOS 27Dec1988
  Microcomputer with priority scheduling

Patents are more than 20 years old and open to public now.



„Scheduler“ uCode PROC ‘s

Scheduler:

- Dequeue
- Run
- StartNextProcess

Links, Timer, Move:

- HandleRunRequest
- HandleReadyRequest
- HandleTimerRequest
- BlockCopyStep
- Insert Step
- Delete Step
- IsThisSelectedProcess

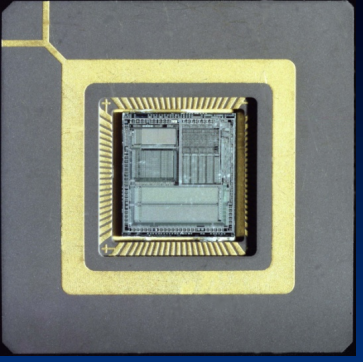
Timer:

- TimeSlice
- InsertInTimerList
- DeleteFromTimerList

Links:

- CauseLinkInput
- CauseLinkOutput
- MakeLinkReadyStatusEnquiry
- EnableLink
- LinkChannelInputAction
- LinkChannelOutputAction

More…?



PROC_Run will start process immediately, if respective queue is empty…

Scheduler uCode Relations

Dequeue

StartNextProcess

not.process

LinkCh Requests

Timer

fetch, decode, execute

j, lend

tin, taltwt

dist

startp, run

stopp, endp

in

out

move

Handl Run Req.

Handle Rdy Req.

Handle Timer Req.

Time Slice

Run

Insert In Timer Queue

Is This Selected Process

Run

SNP:=1

LinkCh Inp Act

LinkCh Ouput Action

Run

.e

.r

.w

INS:=1

SNP:=1

Delete From Timer Queue

DEL:=1

SNP:=1

MOV:=1

Run

SNP:=1

MOV:=1

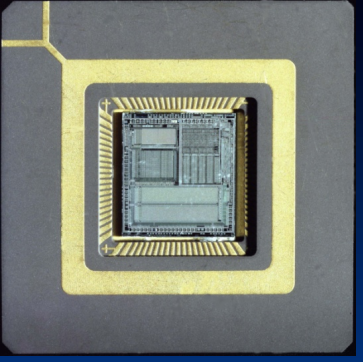
Run

MOV:=1

Run

Run

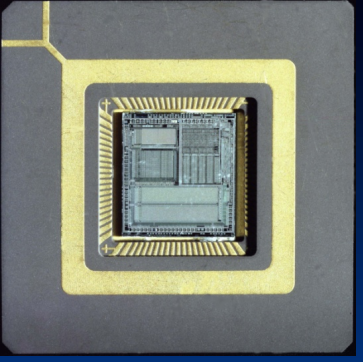
fetch, decode, execute



Open Questions

System Control … reverse engineering tasks:

- Loop bits (DEL, INS, MOVE) for low & high prio ?
- How many HW Status Bits have to be set in parallel ?
- How to distinguish link requests (Ch/run/rdy) ?
- …

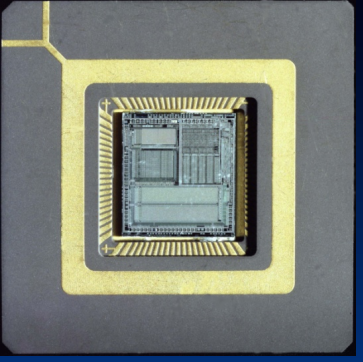


Open Questions

System Control … reverse engineering tasks:

- ✓ Loop bits (low/~~high~~ prio): DEL, INS, COPY, (IOrun)
- 1 How many HW Status Bits have to be set in parallel
- How to distinguish link requests (Ch/run/rdy) → Pat.
- What was the purpose of the DIST SBit ?
- …

Data Path & uCode Refinement … t.b.d.



The most interesting work starts here :

Next Steps … till end 2015

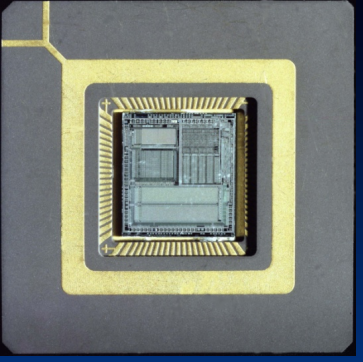
System Control Unit need to be tested and verifyed including:

- Scheduler uCode: StartNextProcess (SNP bit), Dequeue, Run
- OpCodes: in, out, move (MOVE step bit) … in Memory only
- OpCodes: startp, endp, runp, stopp, … alt ……

Timer VHDL to be added…

- Scheduler uCode: Timeslice
- OpCodes: tin (INS step bit), … taltwt … dist (DEL step bit)

Link VHDL … still t.b.d. ……



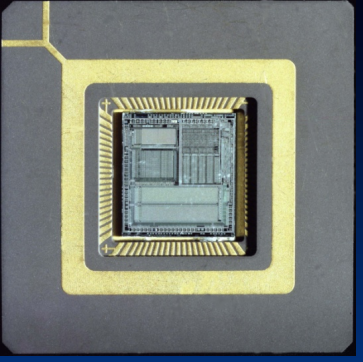
T42 Summary

It can be demonstrated by simulation that …

- Data Path and Control Path (pipeline) are in stable working condition.
- Prefetch Unit and System Control are almost functional, i.e. small instruction sequences can be executed and 32x2k memory can be read and written.

Outlook :

- Scheduler micro-code will become the challenge now, any additional inputs are welcome … Thank You!



Remark

Why there is no project website (www.transputer.eu) yet ?

- Focus on design first and get results = T42 running.
  (maintain documentation, presentation comes later)
- A website is time consuming → permanent effort
  (I ‘m a single person w/ limited physical resources)

PS.: I ‘ve spend a lot of time on Website design trials and a Word Press evening course already …