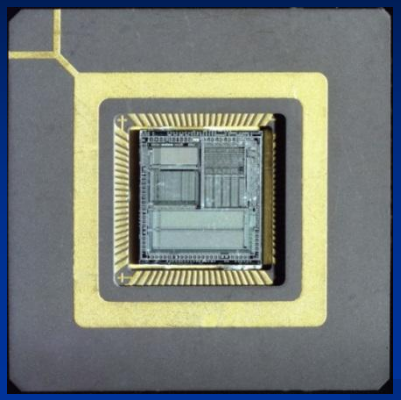


T42 – Transputer Design in FPGA

Year-Two Design Status Report

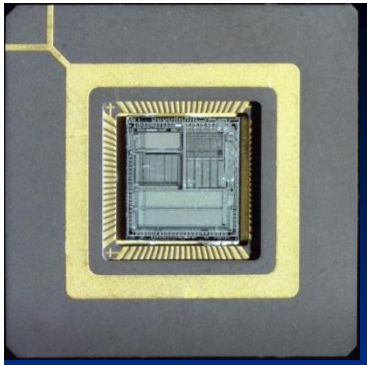
Uwe MIELKE [a] and Martin ZABEL [b],
in collaboration w/ Michael BRUESTLE [c]

[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

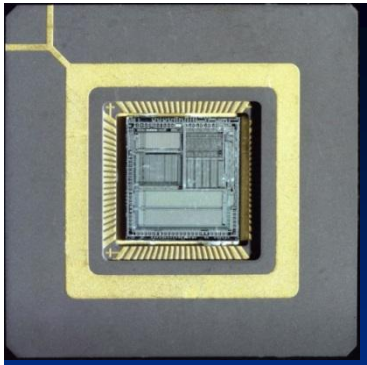
Communicating Process Architectures 2016



T42 in FPGA @ CPA 2016

Abstract:

This fringe session will present the design progress of our IMS-T425 compatible Transputer design in FPGA. The 32bit CPU + Memory interface (2x8kB) are in stable working condition. 117 instructions (from 123+7) are almost implemented in 460 lines of uCode, e.g. TASM loops incl. interruptible MOVE(s) can be simulated some 100 clock cycles. Timer(s) are running. The System control unit allows error mode, MOV-bit and events. Some still open questions around scheduler micro-code and link interaction will be discussed.



Agenda

(1) Achievements (2016 vs 2015)

(2) T42 Schematic Overview

(3) T42 VHDL Top View (2016 vs 2015)

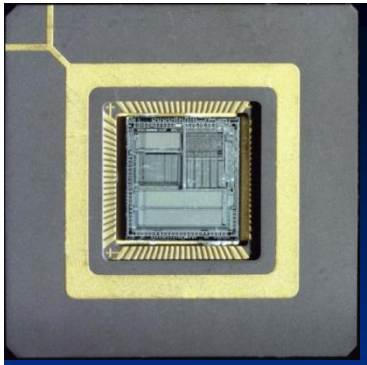
(4) uCode … News

(5) Status Bits for Mov2D

(6) CPU : Cache : DDR-RAM-Ctrl = 1 : 2 : 4

(7) Outlook (2016 vs 2015)

(8) Discussion: Links … (and uCode interaction)



T42 in FPGA @ CPA 2014

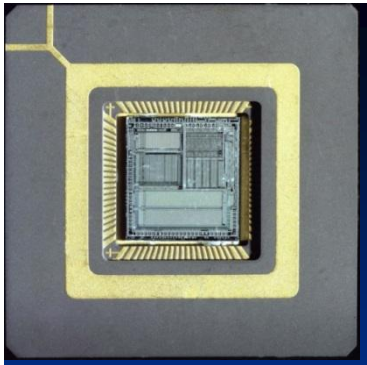
Our Motivation:

- Overcome absence of CSP (Transpu ters and Occam) in public
- Provide a free, IMS -T425 binary compatible, open source VHDL
- Many T42 cores fit into s mall 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 !

old foil from
CPA 2014

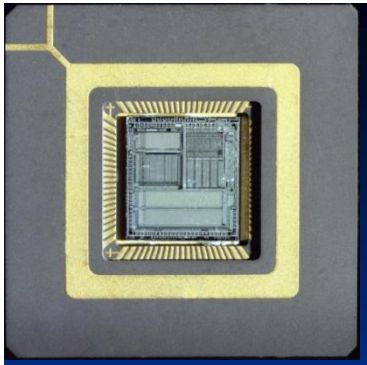


T42 Achievements 2015

- 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, statu s bits, more flags added *… Aug‘2015

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

old foil from
CPA 2015



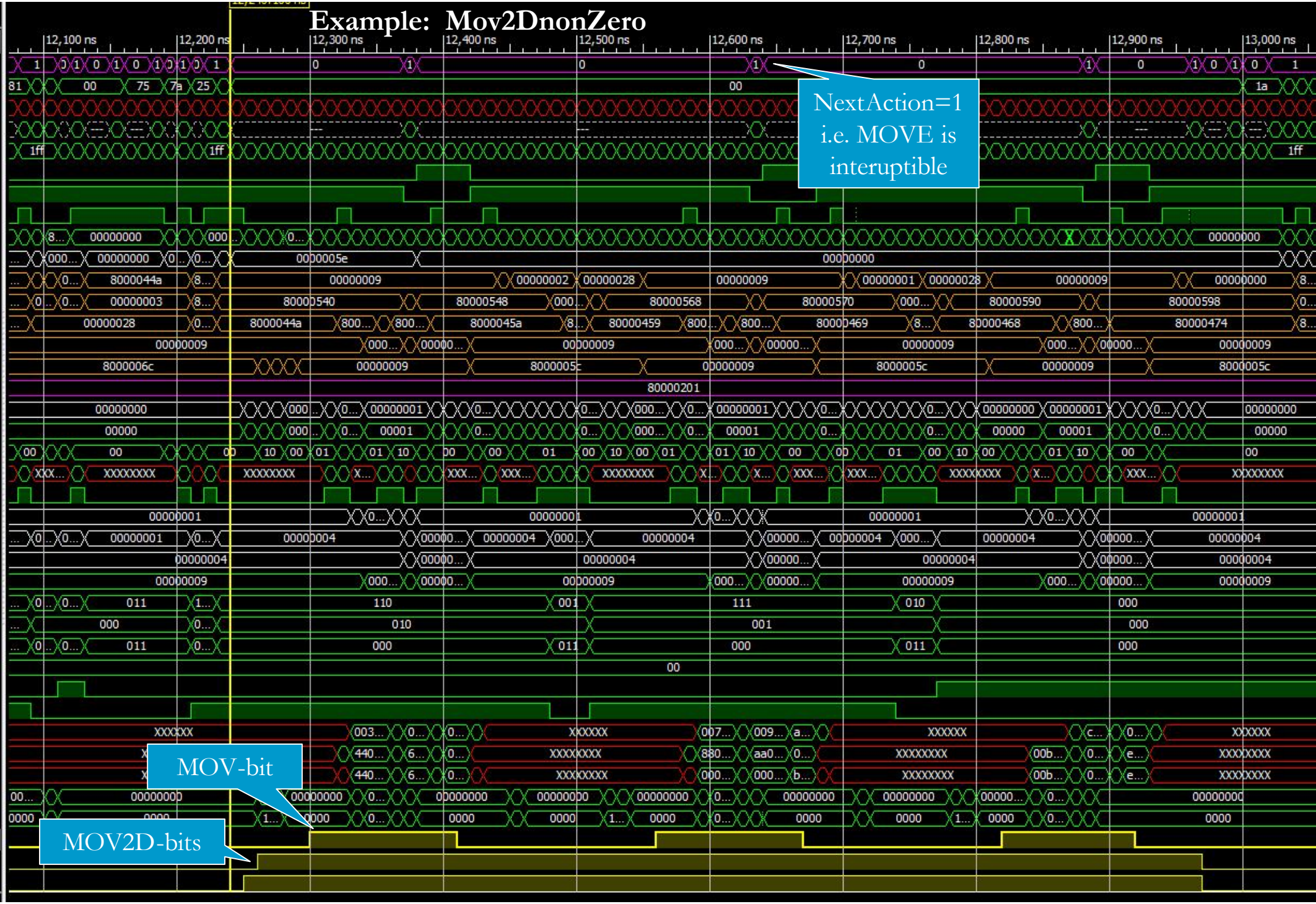
T42 Achievements 2016

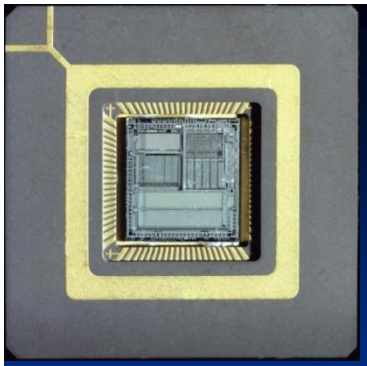
- System control unit, statu s bits to Sreg connected … Aug‘2015
- Timer VHDL (not fully te sted yet, uCode missing !) … Sep‘2015
- Pipelined Oreg within Idecode (hardware Pfix,Nfix) … Nov‘2015
- Move+Move2D: ByteAlign + uCode + Mov -bit Ok… Feb‘2016
- MemIF w/ dual port arbit er completed (8kB + 8kB) … Apr‘2016
- uCode for long Arithmetics, Error Mode tested Ok … May‘2016
- uCode for In, Out, ALT ‘s (no timer ! still ongoing) … Jun‘2016
- Scheduler uCode (some 1st routines, still ongoing) … Jul‘2016
- 1st trial VHDL of (the m ost simple) Output Link … Aug‘2016

Note: >460 lines uCode written (from 512, i.e. uCodeROM is almost full)!

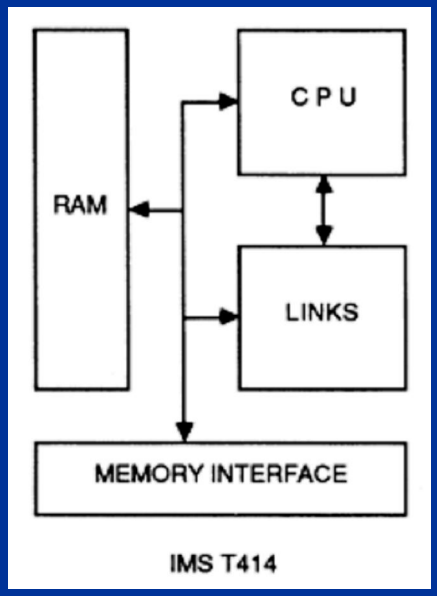
Intension was to understand influence of uCode on DataPath+Sytem structure.

Example: Mov2DnonZero





T42 Schematic 2016



8kB DPRAM

(On-Chip)

2nd

8kB DPRAM

(preliminary instead of Caches)

Fetch and Instr. Bus

Addr and Data Bus

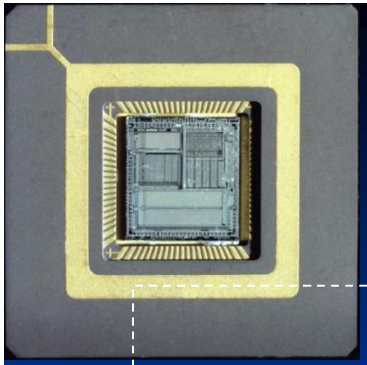
Timers

T42-CPU

512x 96bit uCode ROM

Link 0-3 & DMA's (N/A)

System Services



Remark: Blocks in red still N/A.

T42 VHDL Top View 2015

T42cpu_all_top (structural)

SysPath:
- SysCtrl, 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

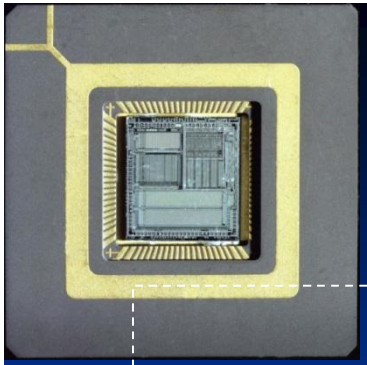
LinkPath:

MemPath:
- MemIF
- MemMain (2kx32)
- DCache
- ICache
- eMemIF

Target Board No.2 199$
Digilent ATLYS
MemDDR2
(64Mx16 on board)
XC6LX45

Target Board No.1 89$
Avnet Micro Board
MemLPDDR
(32Mx16 on board
XC6LX9

old foil from
CPA 2015



Remark: Blocks in red still N/A.

T42 VHDL Top View 2016

T42cpu_all_top (structural)

T42_cpu_constpkg

SysPath:
- SysCtrl, Sbits, Timer, SysService

Ctrl2Data (structural) ← *Pipeline*

CtrlPath:
- ÅuCodeROM
- Idecode
- Oreg *(pipe)*
- Iptr *(+Inc)*
- PreFetch

DataPath:
- ÅABCDEFreg
- ÅALU *X+Y=Z*
- ÅWptr
- ÅPointers
- ÅConstBox
- ÅByteAlign

LinkPath:
ÅSync, ChIn, ChOut, ChEvent, Ifos

MemPath:
- ÅMemIF
- ÅMemMain (dpram2kx32)

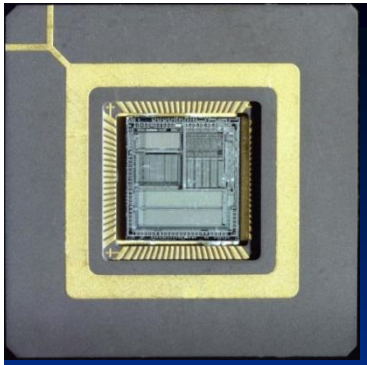
preliminaryéé instead of cache
- ÅDummyCache (dpram2kx32)

available+tested:
- ÅCacheCtrl *(TUD)*
- ÅDDRCtrl *(TUD)*

Target Board No.3 99$
Digilent Arty
MemDDR3
(128Mx16 on board)
XC7A35T

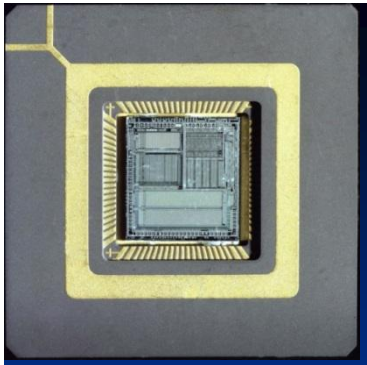
Target Board No.2 199$
Digilent ATLYS
MemDDR2
(64Mx16 on board)
XC6LX45

Target Board No.1 89$
Avnet Micro Board
MemLPDDR
(32Mx16 on board
XC6LX9



T42 uCode … News

- T42: still … 96bit wide (about ~38 signals), more than 460 lines of uCode written up to today!
- T425 (uCodeROM ~60kBit) *seems to be* >100bit wide, having more than 512 uWords … uCode Subroutines?
- T42 w/o call & return stack, i.e. few repetitions in uCode
- *Example* MOVE: 21 uWords + 15 uWords MOV2D
- *Example* OUT: 16 uWords + ?? uWords for Link-HW
- *Example* 11xALT: 73 uWords + ?? uWs. for Timer-HW
- *Example* DIV/REM/LDIV: 1 algo but 3x ~14 uWords



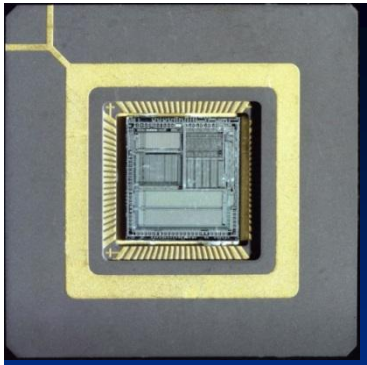
Status Bits for MOV2D (1/2)

Thanks to **Michael Bruestle** for Evaluation (28-Mar-2016)
…due to not available in INMOS documentation so far.

bit31 bit16/15 bit0

E	j0Break	Mov2Dnonzero	Mov2Dall	H	T	I	D	M	R	G	0

```
Sreg(00)              <-- '0'
Sreg(01)              <-- S_Bit(1) GoSNP
Sreg(02)              <-- S_Bit(2) IORun (used by IN, OUT to run Ereg after Move)
Sreg(03)              <-- S_Bit(3) MOV = COPY Flag
Sreg(04)              <-- S_Bit(4) DEL
Sreg(05)              <-- S_Bit(5) INS
Sreg(06)              <-- '0'      DISTandINS ... CPU internal use only:
                                    DISable Timer while INSerting process in timer queue
Sreg(07)              <-- S_Bit(7) HALTonError
Sreg(15 downto 08)    <-- S_Bit(8) 2Dall_Flag (8x) ... MOVE/2D ALL
Sreg(22 downto 16)    <-- S_Bit(9) 2Dnon_Flag (7x) ... MOVE/2D NONZERO
Sreg(30 downto 23)    <-- S_Bit(14) j0Break (8x)
Sreg(31)              <-- S_Bit(15) Error -> Error_out pin <- Error_in pin
```



Status Bits for MOV2D (1/2)

Thanks to **Michael Bruestle** for Evaluation (28-Mar-2016)

Move2D data structure:

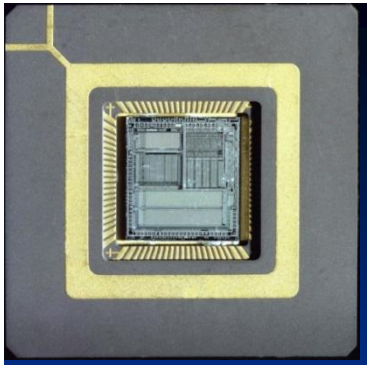
```
for high prio from 0x80000048
-- MinInt +12 to +16
for  low prio from 0x8000005C
-- MinInt +17 to +21

M2D_BLK_LENGTH              0
M2D_DST_POINTER             1
M2D_DST_STRIDE              2
M2D_SRC_POINTER             3
M2D_SRC_STRIDE              4
```

Status bit coding for MOV2D:

Sbit (9) 2Dnon (zero)	SBit (8) 2Dall	SBit (3) MOV	
0	0	1	MOVE
0	1	1	MOVE2DALL
1	0	1	MOVE2DNONZERO
1	1	1	MOVE2DZERO

content of **Ereg**SaveLoc in case of interrupt (IORu n := '1') is:

- MOVE ... WDesc (from IN/OUT) process to run after MOVE final step
- MOVE2D ... initial **Areg** value (byte count per line) for next MOVE-loop



CPU:Cache:RAMCtrl = 1:2:4

Thanks to **Martin Zabel** for Estimations (01-Jul-2016)

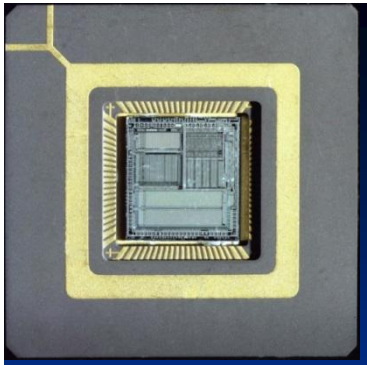
T42 & DDR-RAM	Spartan 6 LUTs / BRAM	Artix 7 LUTs / BRAM
T42 core (16-May-2016 w/o Links) T42 links (estimation)	1800 / 7 1200 .	~same expected~
8kB Cache (16 Byte = 128bit per Line) Controller (4x associative) + Tag RAM	4000 / 4	~same~
8kB Cache (16 Byte = 128bit per Line) Controller (16x associative) + Tag RAM	5100 / 4	~same~
DDR/2/3 Controller (multi bank capable)	Xilinx Hw. MCB + 700 .	7000 * .

FPGA utilization of a minimal configuration (3000+4000+700 = 7700 LUTs):

ÅXC6LX9 (5720 LUTs / 32 BRAMs) LUTs > 100% / BRAMs ~ 34%

ÅXC6LX45 (27228 LUTs / 116 BRAMs) LUTs ~ 28% / BRAMs ~ 9%

ÅXC7AT35 (20568 LUTs / 65 BRAMs) * LUTs > 71% / BRAMs ~ 17%



Open Questions 2015

Done till Aug.2016

System Control Unit need to be tested and verifyed : **50%**

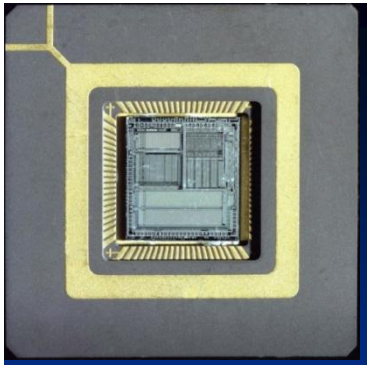
- Scheduler uCode: StartNextProcess, Dequeue, Run **50%**
- OpCodes: in, out, move (MOV -bit) … in Memory only ..... **done**
- OpCodes: startp, endp, runp, stopp, alt ‘s **90%**

Timer VHDL to be added .... **done**

- Scheduler uCode: Timeslice **50%**
- OpCodes: tin, taltwt (INS step bit), dist (DEL step bit) **10%**

Link VHDL still t.b.d. **10%**

old foil from
CPA 2015



The interesting work starts here :

Next Steps till end 2016+

System Control Unit must be completed: **50%**

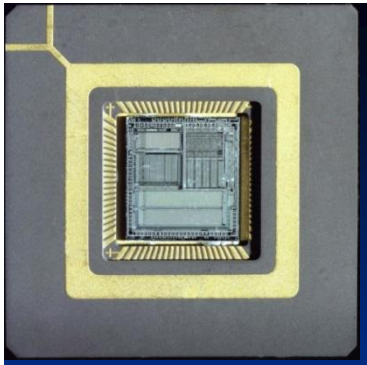
- Scheduler uCode: SNP, Dequeue, Enqueue, Run **50%**
- Analyze (determined stop after descheduling points) ..... t.b.d.

Timer VHDL to be tested ..... t.b.d.

- OpCodes: tin, taltwt (INS step bit), dist (DEL step bit) .....t.b.d.
- Scheduler uCode and HW interaction **50%**

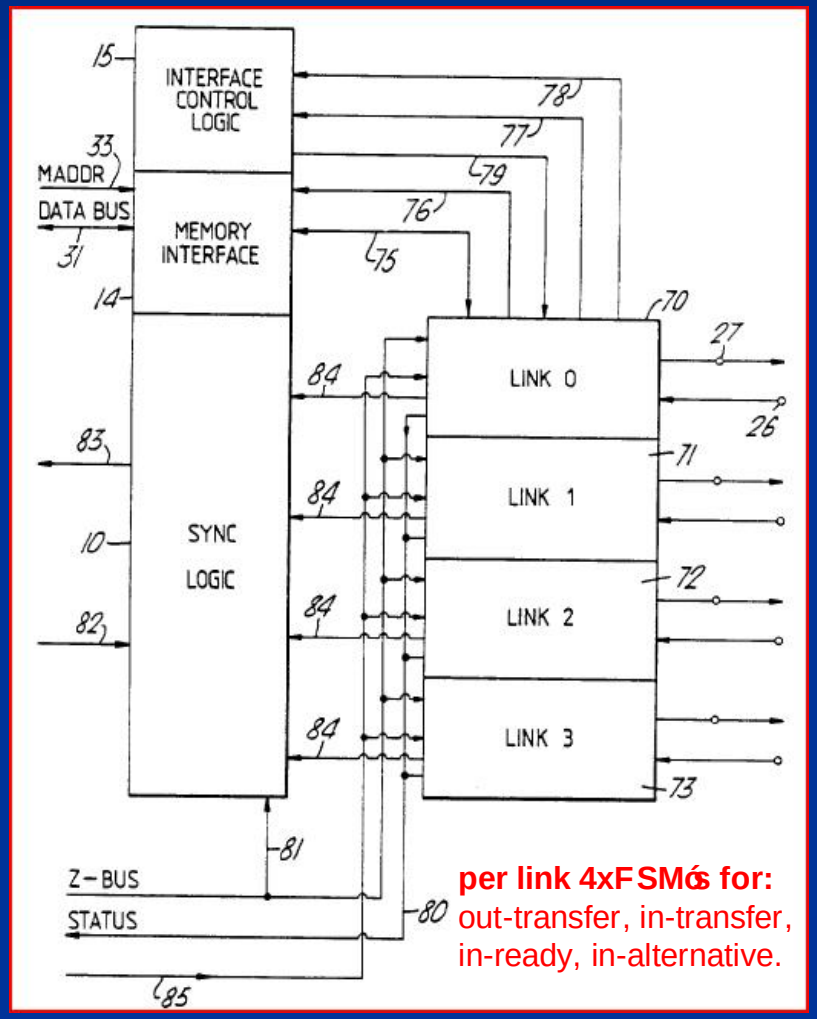
Link VHDL (Out, In, Event) still t.b.d. ....t.b.d.

- Scheduler uCode and HW interaction ....t.b.d.

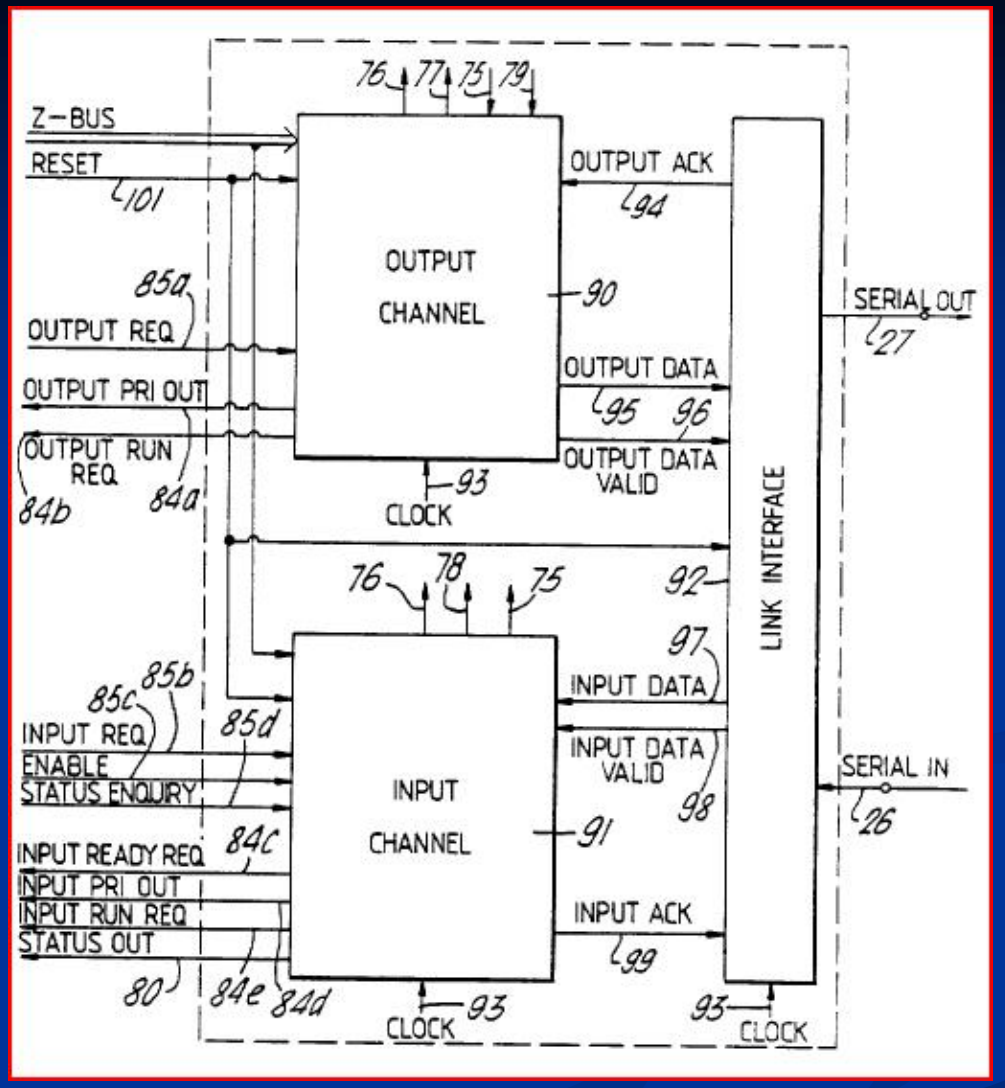


Links …

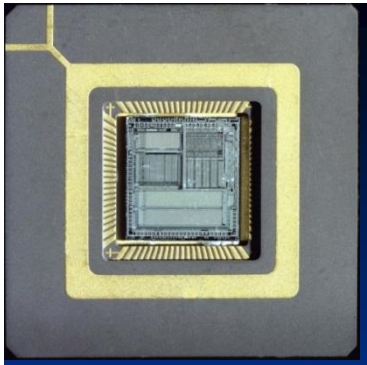
PS.: no "old" VHDL on the internet anymore, my source: Pat-4783734



per link 4xFSMs for:
out-transfer, in-transfer, in-ready, in-alternative.



- Link-Stack: Breg (Zbus) → CountReg → PtrReg → DBuffReg → (Ubus) Areg
- Idea: use (additional) Sbits to give start & stop pulses to the link state machines.

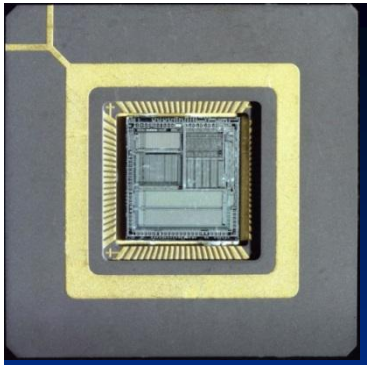


Outlook 2016Q4+

- Reverse engineering required: **Links** + Control Logic
- Final test of … Mul‘s, Div‘s, In, Out, ALT‘s, PAR‘s …
- Scheduler **uCode** completion: HW event interaction … w/ Timer & Links, Boot, Peek, Poke, … Analyze
- Write leftover instructions: Crc‘s, Bitrev‘s, Bitcnt, Unpack, Rounds, Postnorm, … Testhardchan
- uCode ROM increase … and/or Call&Return-Stack ?

Target of all investigations is: getting the full overview!

… and it seems there‘s „not that much more“ leftover …



T42 Summary

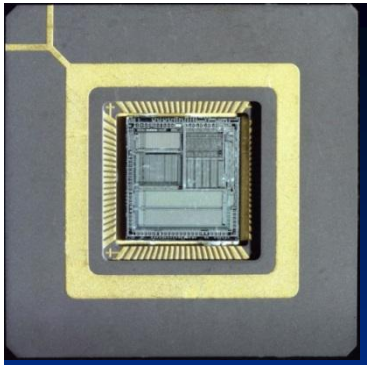
It can be demonstrated by simulation that …

- CPU itself (+ Memory) is in stable working condition, but still has to undergo further refinement.
- System Control Unit has proven its basic functionality, what can be enhanced for still t.b.d. needs.

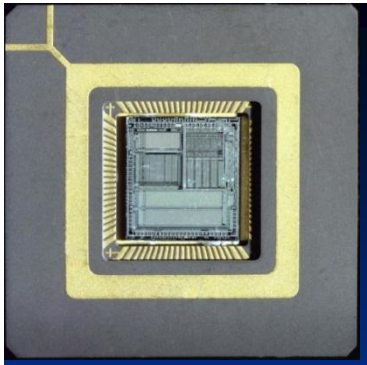
P.S.: simulation of assembler sniplets for some 100 clock's achieved.

Outlook :

- Challenge no.1: Link VHDL incl. FSM's
- Challenge no.2: uCode … Scheduler+HW interactions & optimization of size.



BACKUP

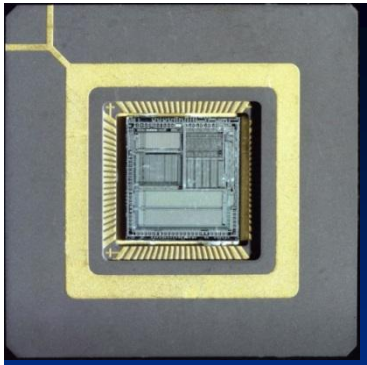


INMOS Patent Research

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 prior ity scheduling

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



www.transputer.eu

Demand & priority for project website is growing, i.e. …

project website preparations are ongoing in background.

Plan is to launch by end 2016 w/ minimalistic content:

- Transputer architecture lessons (HW & ISA)
- brief info about ongoing T42 design project
- inquiry for legacy application source code & lib 's