



Vad är en multicorearkitektur?

Text: Joel Huselius

För en tid sedan (dvs 10 till 40 år sedan, beroende på referensram) var det populärt med multiprocessorsystem. Flera datorer med en processor kopplades ihop via nätverk och programmerades att lösa en gemensam uppgift. Multicore är nästan samma sak, men alla processorer (eller kärnor som de nu kallas) är placerade på samma kiselchip i samma dator. Denna starka integration skapar förutsättning för ytterligare prestandaförbättring, men utgör också grogrund för ytterligare problem.

Bus versus NoC

Den kanske mest naturliga interconnect-lösningen (dvs hur kärnor i ett multicore-chip kopplas ihop) är genom en gemensam bus. Tyvärr visar det sig att chip med fler än tio kärnor förlorar i prestanda med en sådan lösning. Ett alternativ kallas för Network-on-Chip (NoC). Genom att implementera ett internt paketbaserat nätverk på chipet kan komplicerade strukturer och hierarkier av enheter implementeras. Nedan presenterar vi två kommersiella arkitekturer. Vi kommer särskilt att titta på deras olika interconnect-lösningar och hur de olika kärnorna kan samarbeta för att lösa en uppgift.

IBM Cell

IBM har definierat Cell Broadband Engine Architecture (CBEA, eller Cell) som ett ramverk vilket är tänkt att omfatta

VI ÄR ALLA KONSULTER PÅ COMBITECH, ett av Sveriges största konsultföretag som kombinerar teknik, miljö och säkerhet. Vi är ca 800 medarbetare på 20 orter.

flera generationer Cell-processorer. Cell är baserat runt en central bus vid vilken elva enheter är placerade: en reguljär 64-bit Power kärna (PPE), åtta mindre vektorkärnor (SPEs), och två minnesmappade I/O enheter. För framtida utbyggnad stödjer arkitekturen chip-designer med fler PPE- och SPE-enheter. Processorn är primärt tänkt till spel och strömmande Internet-applikationer, till vilket vektorprocessorer med SIMD-arkitekturer visat sig särskilt effektiva. Även PPE:n har visst stöd för SIMD-instruktioner. PPE har två hårdvarutrådar (det vill säga, den kan exekvera två trådar helt parallellt under förutsättning att de bägge trådarna har samma minnesrymd) och är specialiserad för multitrådning och kontrollintensiva uppgifter, medan SPE är specialiserad för beräkningsintensiva uppgifter. Båda enhetstyperna kan dock lösa samtliga uppgifter.

Minnesrymden är delad mellan kärnorna, men varje SPE har även ett eget minne. En DMA-enhet (Direct Memory Access) används för att snabbt och effektivt flytta data mellan det egna och det delade minnet via den gemensamma bussen. Motiveringen till denna lösning är att minska tiden kärnan spenderar åt att vänta på minnesåtkomster. Det är normalt att vektorinstruktioner använder mycket minne och åtkomst till det egna minnet är mer pålitligt, mer tidsdeterministiskt och inte lika tidsödande

att använda som det delade. Om mjukvaran byggs upp som en pipeline, så att DMA-enheten kan överföra ny data medan SPE:n bearbetar tidigare data, så kan uppsnabbningen relativt vanliga processorer bli mycket stor. Det finns möjlighet att konfigurera det lokala minnet för en SPE så att även andra enheter kan komma åt det.

PPE:n, som inte använder vektorinstruktioner i lika hög utsträckning som SPE-enheter, kommunicerar med minnet genom vanliga load/store instruktioner och har både L1 och L2 cache. Cacheminnena avlyssnar den gemensamma bussen för att kunna behålla sin konsistens.

Den centrala bussen (EIB) är utformad som fyra parallella cirkulära bussar för data, den kan hantera fler än 100 utstående DMA-förfrågningar. Varje buss kan hantera tre icke överlappande transfereringar parallellt.

Cell har en mycket stor spridning, och genom att vara en av de första riktigt innovativa multicore-arkitekturerna samtidigt som flera stora företag står bakom satsningen har Cell fått mycket publicitet.

Tile64

Företaget Tiler är sprunget ur forskning från MIT. 2007 släpptes Tile64 som är deras

första produkt. Det finns idag två olika fungerade Linux-alternativ till processorn.

Chipet är uppbyggd som en matris av 8x8 generiska processorblock, s.k. Tiles, och ett antal perifera styrenheter för att kommunicera med minne och dyl. Alla Tiles ser likadana ut, de har tre skalärer, egen interrupthantering, DMA-hantering, minnesskydd, L1 cache, L2 cache och virtuellt minne. Tiles delar en gemensam L3 cache som (maximalt) är aggregatet av alla L2 cacheminnen på chipet.

Interconnect-lösningen, iMesh, är generisk och integrerad i Tile-designen, så interconnect-komplexiteten är inte beroende av antalet Tiles. När en Tile vill kommunicera med Tiles eller styrenheter som den inte är direkt ansluten till så routas meddelanden via iMesh-enheter i mellanliggande Tiles. Detta kräver ingen overhead hos dessa indirekt inblandade Tiles, men kan medföra visst jitter på grund av bandbreddskonkurrens.

Implementationen av iMesh är byggd runt synen att eftersom kommunikationsledningar i kisel-design ofta kan lagras ovanpå annan implementation så tar logik mer plats än kommunikationsledningar. Därför är det mer platseffektivt att implementera flera parallella fysiska kanaler än att skapa flera virtuella kanaler i en fysisk kanal. Fem separata nätverk med olika egenskaper och

ansvarsområden har implementerats i Tile64. Tre av dessa är explicit styrda av mjukvara genom system I/O och skalär respektive strömmande inter-Tile-kommunikation. De två nätverk som inte styrs direkt av mjukvaruexekvering sköter minnesreferenser mellan Tiles för att stödja den distribuerade cachen, respektive minnesadministration vid cachemissar, DMA och liknande. Vid strömmande inter-Tile-kommunikation sätts routing upp programmatiskt, medan de övriga fyra nätverken är automatiskt routade. Det är möjligt stänga av utgående kommunikation i de tre mjukvarustyrda nätverken för varje Tile för att förhindra påverkan mellan delsystem; varje delsystem kan sedan köra en egen instans av Linux.

Tile64 har inte rekord i antal kärnor, det finns ett fåtal mindre aktörer som har chip med fler än tvåhundra kärnor, men Tiler har stått för stort nytänkande och även publicerat detaljer och motiveringar till flera av sina lösningar.

Hur ser framtiden ut?

Hur många kärnor processorerna i systemet närmast dig har om tre år beror mycket på systemets problemdomän. Vissa problem är möjliga att parallellisera hårt, medan andra bara lite grann och somliga inte alls. Det beror även på vilka programmeringsmodeller som framtiden erbjuder. Genom t ex virtualisering kan det

vara möjligt att exekvera en semiparallell implementation på en underliggande parallell arkitektur.

Att jämföra Cell och Tile64 är intressant eftersom de har radikalt olika lösningar för de gemensamma problem som alla multicore-arkitekturer står inför. Även om det finns mycket att lära från tidigare multiprocessor-lösningar så är det många val som måste göras och många nya lösningar som kommer att uppträffa. Det kommer att ta lång tid för den här marknaden att svänga in, varken Cell eller Tile64 har alla svar.

Oavsett hur kärnor kopplas ihop i framtiden kommer multicore att vara tillräckligt vanligt för att förtjäna uppmärksamhet idag. Titta på datorn som står på ditt skrivbord idag – finns där en etikett där det står "Core 2 Duo"? Då använder du redan multicore!

Not. Dr. Joel Huselius driver ett lokalt nätverk i Älvsjö för att fördjupa Combitechs kunskap om mjukvaruutveckling för multicore.



VI ÄR ALLA KONSULTER PÅ COMBITECH, ett av Sveriges största konsultföretag som kombinerar teknik, miljö och säkerhet. Vi är ca 800 medarbetare på 20 orter.